ΕΘΝΙΚΌ ΜΕΤΣΌΒΙΟ ΠΟΛΥΤΕΧΝΕΊΟ
ΣΧΟΛΉ ΗΛΕΚΤΡΟΛΌΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΈΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Ανάπτυξη Προσομοιωτή Δικτυακής Κίνησης σε Νοσοκομειακά Περιβάλλοντα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Λουκάς Π. Τουρίκης

Επιβλέπων : Δημήτρης Ασκούνης
Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2021

ΕΘΝΙΚΌ ΜΕΤΣΌΒΙΟ ΠΟΛΥΤΕΧΝΕΊΟ
ΣΧΟΛΉ ΗΛΕΚΤΡΟΛΌΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΈΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Ανάπτυξη Προσομοιωτή Δικτυακής Κίνησης σε Νοσοκομειακά Περιβάλλοντα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Λουκάς Π. Τουρίκης

Επιβλέπων : Δημήτρης Ασκούνης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 5^{Η} Νοεμβρίου 2021

.....	.....	.....
Δημήτρης Ασκούνης	Ιωάννης Ψαρράς	Χρυσόστομος Δούκας
Καθηγητής ΕΜΠ	Καθηγητής ΕΜΠ	Αναπληρωτής Καθηγητής ΕΜΠ

Αθήνα, Νοέμβριος 2021

.....

Λουκάς Π. Τουρίκης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Ο τομέας της υγείας αντιμετωπίζει έναν μεγάλο αριθμό ζητημάτων κυβερνοασφάλειας. Τα ζητήματα αυτά κυμαίνονται από επιθέσεις κακόβουλου λογισμικού με στόχο τα προσωπικά δεδομένα ασθενών έως επιθέσεις που έχουν ως αποτέλεσμα την αδυναμία περίθαλψης ασθενών και θέτουν σε κίνδυνο τους τρεις πυλώνες των ιατρικών δεδομένων: εχεμύθεια, ακεραιότητα, διαθεσιμότητα. Για τα συστήματα υγείας, οι κυβερνοεπιθέσεις μπορούν να έχουν επιπτώσεις και πέραν των οικονομικών ζημιών και την ασφάλεια των προσωπικών δεδομένων.

Η ανίχνευση δικτυακών επιθέσεων είναι ένα από τα σημαντικότερα προβλήματα για την ασφάλεια των σύγχρονων δικτύων. Για την ανάπτυξη αποτελεσματικών μεθόδων ανίνευσης εισβολής, είναι απαραίτητα ρεαλιστικά και σύγχρονα σετ δεδομένων. Το πεδίο έρευνας της παρούσης διπλωματικής εργασίας αφορά την ανάπτυξη ενός προσομοιώτη δικτυακής κίνησης νοσοκομειακών εγκαταστάσεων. Συγκεκριμένα, προσομοιώνει την ψηφιακή συμπεριφορά του ιατρικού και μη προσωπικού ενός νοσοκομείου, με απώτερο σκοπό την παραγωγή καθημερινής καλόβουλης δικτυακής κίνησης. Η εφαρμογή επιτρέπει την δημιουργία προφίλ χρηστών, τα οποία είναι παραμετροποιήσιμα ώστε να προσομοιώνονται ρεαλιστικά οι ρόλοι του γιατρού, του διοικητικού προσωπικού κ.α.

Λέξεις Κλειδιά

Κυβερνοασφάλεια, Συμπεριφορά Χρηστών, Προσομοίωση, Ανάλυση Δεδομένων

Abstract

The healthcare industry is plagued by a large number of cybersecurity related issues. These issues can range from malware that compromises the integrity of systems and privacy of patients, to hack attacks that disrupt facilities' ability to provide proper patient care and even jeopardise the three attributes of healthcare information: confidentiality, integrity, availability. For healthcare, cyber attacks can have ramifications beyond financial damages and breach of privacy.

Network intrusion detection is one of the main problems in ensuring the security of modern networks. For the development of efficient network intrusion detection methods, realistic and up to date network traffic datasets are required. The research field of this thesis concerns the development of a network behaviour simulation tool that produces network traffic of hospital facilities. Specifically, the aforementioned tool simulates the digital behaviour, of the health and non-health (IT & secretariat) staff of the hospital with the ultimate goal of creating benign everyday network hospital traffic. The tool allows the creation of user profiles that are configurable with respect to the users' online behavior so that they can realistically simulate client roles such as doctor, administrative staff, etc.

Keywords

Cybersecurity, User Behaviour, Simulation, Emulation, Data Analysis, User Profiling, Device Profiling

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στον τομέα Ηλεκτρικών Βιομηχανικών Διατάξεων και Συστημάτων Αποφάσεων της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών ΕΜΠ. Θα ήθελα αρχικά να ευχαριστήσω τον επιβλέπων καθηγητή κ. Δημήτρη Ασκούνη για την ευκαιρία που μου έδωσε να εκπονήσω την παρούσα εργασία πάνω σε ένα τόσο ενδιαφέρον θέμα. Επίσης θα ήθελα να ευχαριστήσω ιδιαίτερα τους Υποψήφιους Διδάκτορες ΕΜΠ κ. Σωτήρη Πελέκη και κ. Μιχαήλ Κοντούλη για την άψογη συνεργασία και πολύτιμη καθοδήγηση τους, χωρίς την βοήθεια και επιμέλεια των οποίων δεν θα μπορούσε να ολοκληρωθεί η παρούσα διπλωματική. Κλείνοντας δεν θα μπορούσα να παραλείψω τις ευχαριστίες στην οικογένεια μου, που με στήριξε όλα αυτά τα χρόνια σπουδών και ήταν δίπλα μου σε κάθε δυσκολία.

Περιεχόμενα

Ευρετήριο Εικόνων

Ευρετήριο Πινάκων

Κεφάλαιο 1

Εισαγωγή

1.1 Αντικείμενο - Σκοπός

Τα τελευταία χρόνια είναι γεγονός η ραγδαία αύξηση των κυβερνοεπιθέσεων με εμφανείς επιπτώσεις στην οικονομία και στην κοινωνία γενικότερα. Συχνό πλέον στόχο αποτελούν κρίσιμες εγκαταστάσεις και συστήματα του κλάδου της υγείας και κατ' επέκταση νοσοκομειακά δίκτυα και οργανισμοί. Οι συνέπειες τέτοιων επιθέσεων κυμαίνονται από το να μένουν ολόκληρα νοσοκομεία χωρίς τη δυνατότητα χρήσης των πληροφοριακών τους συστημάτων, έως και ακόμη, τον θάνατο ασθενών.

Για την αντιμετώπιση των επιθέσεων αυτών έχουν αναπτυχθεί μια σειρά από εργαλεία, με έμφαση σε τεχνικές μηχανικής μάθησης (machine learning) και ανίχνευσης ανωμαλιών. Η εκπαίδευση αυτών των εργαλείων, ώστε να μπορούν να ανιχνεύσουν αποτελεσματικά κυβερνοεπιθέσεις, γίνεται με την χρήση σετ δεδομένων (datasets) δικτυακής κίνησης. Η χρήση όμως σετ δεδομένων που προέρχονται από πραγματική δικτυακή κίνηση καθίστανται απαγορευτική για λόγους προστασίας προσωπικών δεδομένων και άρα είναι αναγκαιότητα η δημιουργία νέων συνθετικών σετ δεδομένων. Η παραγωγή των τελευταίων επιτυγχάνεται από αντίστοιχους προσομοιωτές.

Αντικείμενο της παρούσας διπλωματικής, είναι η ανάπτυξη προσομοιωτή της δικτυακής συμπεριφοράς του προσωπικού (και κατ' επέκταση της δικτυακής κίνησης) ενός νοσοκομείου. Η τελευταία συμπεριλαμβάνει μεταξύ άλλων δικτυακές συμπεριφορές όπως web browsing, chatting, ανταλλαγή μηνυμάτων ηλεκτρονικού ταχυδρομείου, αλλά και αλληλεπίδραση μεταξύ του προσωπικού και των ενδονοσοκομειακών server και συστημάτων, στο πλαίσιο της αναπαραγωγής στον μεγαλύτερο δυνατό βαθμό, ρεαλιστικής και σε αντιστοίχιση με την πραγματική κίνηση ενός νοσοκομειακού δικτύου. Η κίνηση που αναπαράγεται μπορεί να χρησιμοποιηθεί και για την εκπαίδευση εργαλείων σε ευρύτερους κλάδους, αν και κρίνεται ως πλέον κατάλληλη λόγω των ιδιαίτερων χαρακτηριστικών της, για την εκπαίδευση αλγορίθμων που αναπτύσσονται για την προστασία νοσοκομειακών εγκαταστάσεων.

1.2 Οργάνωση του τόμου

Η διπλωματική εργασία είναι οργανωμένη σε πέντε κεφάλαια. Η δομή τους είναι η εξής:

Το πρώτο κεφάλαιο είναι η Εισαγωγή. Στο δεύτερο κεφάλαιο παρουσιάζεται το θεωρητικό υπόβαθρο για την ανάπτυξη του Behaviour Simulator που περιλαμβάνει σύντομες παρουσιάσεις των συστημάτων πρόληψης και ανίχνευσης εισβολής (IPS και IDS αντίστοιχα), των ήδη διαθέσιμων σετ δεδομένων καθώς και τις απαιτήσεις για ένα σύγχρονο σετ δεδομένων, και ακολουθεί παρουσίαση των εννοιών προφίλ και flow. Το τρίτο κεφάλαιο αφορά την μεθοδολογία της προσομοίωσης, το οποίο περιλαμβάνει ανάλυση της καταγραφής και παρατήρησης των flows, και παρουσιάζει δύο μεθόδους μοντελοποίησης της δικτυακής κίνησης των προφίλ: την χρήση ιστοσελίδων με στατιστικά για την πλοήγηση χρηστών στο διαδίκτυο και την στατιστική ανάλυση πραγματικής δικτυακής κίνησης. Στο τέταρτο κεφάλαιο παρουσιάζονται τα εργαλεία και οι τεχνολογίες που αξιοποιήθηκαν για την ανάπτυξη του προσομοιωτή και τέλος παρουσιάζεται η ίδια η εφαρμογή, με αναλυτική περιγραφή του χειρισμού της μέσω του Swagger RESTful API που αναπτύχθηκε, και τέλος οι λειτουργικότητες της. Το πέμπτο και τελευταίο κεφάλαιο περιλαμβάνει μια σύνοψη και μελλοντικές επεκτάσεις.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

2.1 Οι κυβερνοεπιθέσεις στα νοσοκομειακά δίκτυα

2.1.1 Κυβερνοεπιθέσεις – Ορισμός

Η κυβερνοεπίθεση (αγγλικά: Cyberattack) είναι οποιοδήποτε σύνολο κακόβουλων ενεργειών που λαμβάνει μέρος μέσω ενός ηλεκτρονικού υπολογιστή ή δικτύου, με σκοπό την τροποποίηση, κλοπή, υποκλοπή ή και την μη εξουσιοδοτημένη πρόσβαση στις πληροφορίες του νόμιμου κατόχου.[4] Στόχος κυβερνοεπιθέσεων μπορεί να αποτελέσουν ένας κοινός προσωπικός ηλεκτρονικός υπολογιστής, ένα πληροφοριακό σύστημα, ένα δίκτυο υπολογιστών ακόμα και ολόκληροι οργανισμοί και κράτη.[5] Μια κυβερνοεπίθεση μπορεί να προέλθει επίσης, από ένα κράτος, μια ομάδα, ένα κοινωνικό σύνολο, έναν οργανισμό, μια ανώνυμη πηγή κ.α.[6]

2.1.2 Οι κυβερνοεπιθέσεις στα συστήματα υγείας

Τα συστήματα υγείας αποτελούν όλο και πιο συχνά στόχο κυβερνοεπιθέσεων. [7]
Οι λόγοι του παραπάνω φαινομένου θα μπορούσαν να συμπυκνωθούν στους παρακάτω:

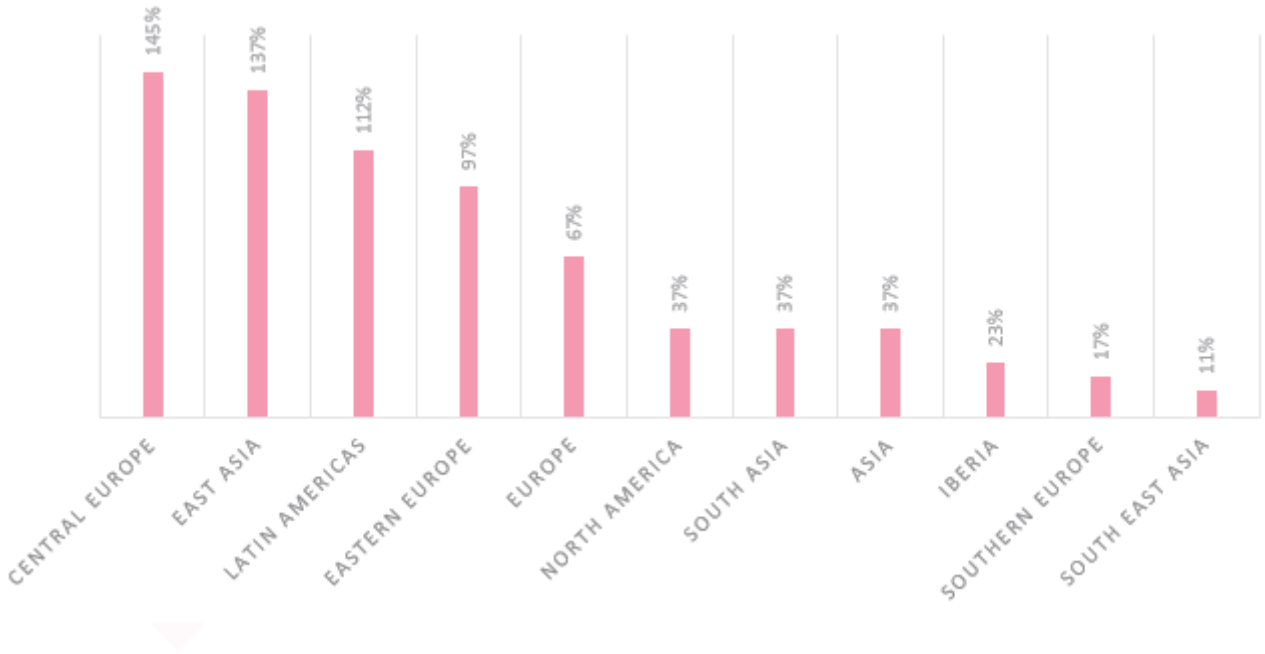
- Οι κυβερνοεγκληματίες θεωρούν τους υγειονομικούς οργανισμούς πιο «πρόθυμους» να ανταποκριθούν στις απαιτήσεις και να πληρώσουν λύτρα.
- Στα νοσοκομεία συγκεντρώνονται μεγάλο πλήθος προσωπικών δεδομένων ασθενών.
- Οι νοσοκομειακές συσκευές δεν έχουν ικανοποιητικές προδιαγραφές ασφαλείας.
- Το προσωπικό πολλές φορές αναγκάζεται να έχει απομακρυσμένη πρόσβαση σε προσωπικά δεδομένα κάτι που εγκυμονεί κινδύνους.
- Παρατηρείται δυσκολία στην μετάβαση σε πιο ασφαλείς πρακτικές και πρωτοκόλλα από τους οργανισμούς υγείας.
- Ελλιπής εκπαίδευση του προσωπικού σε ζητήματα κυβερνοασφάλειας και ασφαλών πρακτικών. [8] [9]

2.1.3 Η επίδραση της πανδημίας του κορονοϊού covid19

Η πανδημία του κορονωιού – covid19 έπαιξε καταλυτικό ρόλο στην περαιτέρω επιδείνωση του φαινομένου. Μετά την έξαρση της πανδημίας παρατηρείται αύξηση στις κυβερνοεπιθέσεις σε οργανισμούς υγειονομικής περίθαλψης παγκοσμίως, καθιστώντας τους οργανισμούς υγειονομικής περίθαλψης πρώτους σε κυβερνοεπιθέσεις. Τα νοσοκομεία είναι ιδιαίτερα ελκυστικοί στόχοι επίθεσης, επειδή είναι πιο ευάλωτα στις ransomware επιθέσεις, εξαιτίας και της μεγάλης πίεσης από τον αναπτυσσόμενο αριθμό περιπτώσεων κορωνοϊού και του προγράμματος εμβολιασμού. Συγκεκριμένα καταγράφηκε αύξηση 45% στις παγκόσμιες κυβερνοεπιθέσεις στον τομέα της υγείας, διπλάσια από την αύξηση των κυβερνοεπιθέσεων σε όλους τους άλλους οργανισμούς (+22%). [10]

Οι αυξήσεις των κυβερνοεπιθέσεων σε οργανισμούς υγειονομικής περίθαλψης σημειώθηκαν κατά κύριο λόγο στην Κεντρική Ευρώπη (+ 145%), ενώ ακολούθησαν, η Ανατολική Ασία (+ 137%), η Λατινική Αμερική (+ 112%), η Ευρώπη (67%) και η Βόρεια Αμερική (37%) . Πιο χαρακτηριστικά, σε συγκεκριμένες χώρες, όπως ο Καναδάς, παρατηρήθηκε αύξηση της τάξεως του 250%, η Γερμανία σημείωσε αύξηση 220% ενώ η

Ισπανία κατέγραψε διπλάσιο αριθμό κυβερνοεπιθέσεων (τύπου ransomware) στον τομέα της υγειονομικής περίθαλψης. [10]



Πίνακας 2.1 Αύξηση επιθέσεων, ανά περιοχή μετά την έξαρση της πανδημίας του κορονοϊού covid19 [1]

2.2 Συστήματα πρόληψης εισβολής (IPS) - Συστήματα ανίχνευσης εισβολής (IDS)

Η εκθετική αύξηση και ανάπτυξη των δικτύων και των εφαρμογών τους, κάνει αυταπόδεικτο και το μέγεθος των δυνητικών καταστροφών και ζημιών, κυρίως σε οικονομικό επίπεδο που μπορούν να προέλθουν από κυβερνοεπιθέσεις. Κάποια από τα πιο σημαντικά εργαλεία ασφάλειας σε τέτοιες επιθέσεις είναι τα συστήματα ανίχνευσης εισβολής (αγγλικά: Intrusion Detection Systems, στο εξής IDS) και τα συστήματα πρόληψης εισβολής (αγγλικά: Intrusion Prevention Systems, στο εξής IPS).

2.2.1 IPS

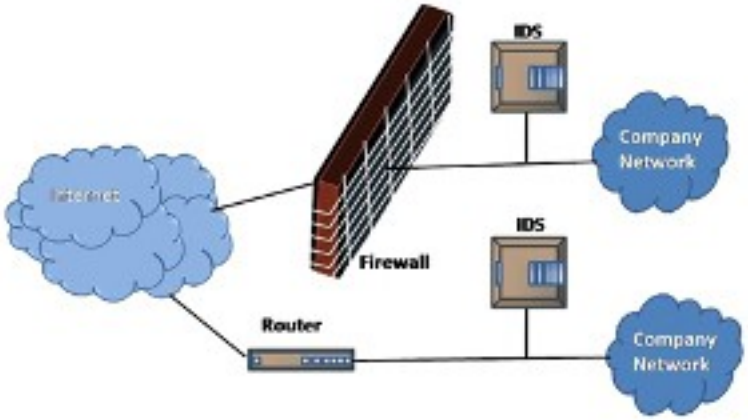
Το IPS είναι μια μορφή ασφάλειας δικτύου που βασίζεται στον έγκαιρο εντοπισμό και άρα την πρόληψη γνωστών επιθέσεων. Ένα τέτοιο σύστημα παρακολουθεί συνεχώς το δίκτυο και αναζητά με διάφορες τεχνικές (δεν ανήκουν στο αντίκειμενο της παρούσης εργασίας, και δεν θα αναλυθούν περαιτέρω) κακόβουλα συμβάντα όπως επιθέσεις άρνησης υπηρεσίας (DoS), κατανεμημένες επιθέσεις άρνησης υπηρεσίας (DdoS), σκουλήκια (worms), ιούς (viruses) κ.α. Στη συνέχεια αναφέρει τα συμβάντα αυτά στους διαχειριστές του συστήματος και παράλληλα λαμβάνει προληπτικά και αποτρεπτικά μέτρα, όπως το

1 [Πηγή: https://blog.checkpoint.com/2021/01/05/attacks-targeting-healthcare-organizations-spike-globally-as-covid-19-cases-rise-again/]

κλείσιμο των σημείων πρόσβασης και την αντίστοιχη διαμόρφωση του τείχους προστασίας (αγγλικά: firewall) για την αποτροπή παρόμοιων μελλοντικών επιθέσεων.[11]

2.2.2 IDS

Τα συστήματα IDS συλλέγουν δεδομένα από μεμονωμένους υπολογιστές ή και από δίκτυα υπολογιστών και τα αναλύουν. Βασικός τους στόχος είναι η έγκαιρη ανίχνευση των επιθέσεων, η κατάλληλη αντιμετώπιση τους και η ενημέρωση του συστήματος και των διαχειριστών του δικτύου.[12] Διακρίνονται σε αυτά που αναγνωρίζουν κακόβουλες επιθέσεις με βάση τις υπογραφές των επιθέσεων (signature based) και σε αυτά που αναγνωρίζουν επιθέσεις με βάση την δικτυακή συμπεριφορά (anomaly based). [13] Τα μεν πρώτα ανιχνεύουν μια επίθεση, όταν αυτή έχει ήδη εκδηλωθεί, συγκρίνοντας την δικτυακή κίνηση με τις υπογραφές επιθέσεων που βρίσκονται αποθηκευμένες σε μία βάση δεδομένων, ενώ τα δεύτερα είναι σχεδιασμένα ώστε να αναγνωρίζουν ανωμαλίες και αποκλίσεις από την κανονική δικτυακή συμπεριφορά, καθώς παρακολουθούν την δικτυακή κίνηση σε πραγματικό χρόνο.[14] Παραδείγματα signature based IDS συστημάτων είναι τα Suricata[2], Snort[3], ενώ αντίστοιχα των anomaly based είναι τα Hogzilla - IDS[4] και Zeek[5]. Στην παρούσα εργασία θα μας απασχολήσουν μόνο τα anomaly based και στο εξής με τον όρο IDS θα αναφερόμαστε σε αυτά.



Εικόνα 2.1 Τοπολογία δικτύου με IDS[6]

2.2.2.1 Χειρισμός επίθεσης από IDS

Η διαδικασία με την οποία ένα IDS χειρίζεται μια επίθεση χωρίζεται σε έξι στάδια:

1. Το πρώτο στάδιο προηγείται της επίθεσης και αποτελεί την προετοιμασία για την επίθεση. Περιλαμβάνει τις διαδικασίες και την εγκατάσταση μηχανισμών για τον εντοπισμό και την απόκριση στην επίθεση.
2. Ακολουθεί η ταυτοποίηση της επίθεσης.

2 [Πηγή: https://suricata.io/]
3 [Πηγή: https://www.snort.org/]
4 [Πηγή: https://ids-hogzilla.org/]
5 [Πηγή: https://zeek.org/]
6 [Πηγή: http://idsportal.cs.teiath.gr/index.php/el/genika/ids]

3. Τρίτο στάδιο είναι ο περιορισμός της επίθεσης, όπου ουσιαστικά περιορίζεται ή και αποκόπτεται η πρόσβαση στο σύστημα για την περαιτέρω επέκταση της ζημιάς. Ταυτόχρονα η επίθεση παρακολουθείται παθητικά και γίνεται καταγραφή των ενεργειών του επιτιθέμενου.
4. Τέταρτο στάδιο είναι η εξουδετέρωση της επίθεσης όπου και διακόπτεται η επίθεση.
5. Ακολουθεί το στάδιο της αποκατάστασης, όπου το σύστημα με βάση την πολιτική ασφαλείας που ακολουθείται επανέρχεται σε μία ασφαλή κατάσταση.
6. Τέλος, παρακολουθείται από θέση ασφαλείας πλέον η επίθεση ώστε να ληφθούν μέτρα κατά του επιτιθέμενου ενώ παράλληλα καταγράφονται τυχόν προβλήματα που προκλήθηκαν όπως και σχετικές εμπειρίες που αποκτήθηκαν. [12]

2.2.2.2 Στόχοι των IDS

Ο κύριος στόχος που οφείλει να ικανοποιεί ένα IDS είναι η αποτελεσματική ανίχνευση όσο το δυνατόν μεγαλύτερου εύρους εισβολών. Αυτό προϋποθέτει να είναι σε θέση να αντιδρά σε αλλαγές στη συνηθή δραστηριότητα του χρήστη και να ανιχνεύει, αν όχι σε πραγματικό χρόνο, σε εύλογα σύντομο χρονικό διάστημα, νέους τύπους επιθέσεων. Στη συνέχεια πρέπει να παρουσιάζει την ανάλυση στο διαχειριστή του συστήματος ώστε ο τελευταίος να λάβει έγκαιρα τα απαραίτητα μέτρα. Ιδιαίτερα σημαντικό είναι τα ποσοστά ψευδών θετικών συναγερμών (αγγλικά: false positives) – όταν αναφέρεται μια επίθεση χωρίς να υπαχει επίθεση σε εξέλιξη, και ψευδών αρνητικών συναγερμών (αγγλικά: false negatives) - όταν δεν αναφέρεται μια επίθεση που είναι σε εξέλιξη, να είναι ελαχιστοποιημένα. [12]

2.2.2.3 Επιθυμητά χαρακτηριστικά των IDS

Τα επιθυμητά χαρακτηριστικά των IDS μπορούν να συνοψίστουν στα εξής:

- Πρώτον, πρέπει να έχουν την ικανότητα παρακολούθησης μεγάλου όγκου δεδομένων.
- Δεύτερον, η ειδοποίηση για την ανακάλυψη μίας επίθεσης από κακόβουλους χρήστες εντός ή εκτός δικτύου, θα πρέπει να γίνεται στο μικρότερο δυνατό χρονικό διάστημα.
- Τρίτον, τα IDS πρέπει να έχουν τη δυνατότητα επέκτασης και τροποποίησης ώστε κάθε φορά να προστατεύουν πιο αποτελεσματικά το σύστημα που έχουν αναλάβει.
- Τέταρτον, θα πρέπει να κάνουν οικονομία ως προς την χρήση των πόρων του συστήματος.
- Τέλος, πρέπει να είναι ανθεκτικά σε επιθέσεις που στοχεύουν στα ίδια.

2.2.2.4 Εκπαίδευση των IDS με χρήση σετ δεδομένων (dataset)

Ο ραγδαία αναπτυσσόμενος αριθμός απειλών ασφαλείας στο ίντερνετ και στα ενδοδίκτυα οργανισμών (αγγλικά: intranets), έχουν κάνει πλέον επιτακτική την ανάγκη για αξιόπιστα συστήματα ασφαλείας. Η αξιολόγηση των συστημάτων ασφαλείας IDS γίνεται με την χρήση σετ δεδομένων. [3] Πολλά από τα ήδη παραχθέντα σετ δεδομένων, δεν είναι διαθέσιμα στο ευρύ κοινό και την επιστημονική κοινότητα για λόγους προστασίας προσωπικών δεδομένων, ενώ σε άλλα, για παρόμοιους λόγους απουσιάζει σημαντικό τμήμα από τα δεδομένα τους με αποτέλεσμα να υστερούν σε συγκεκριμένα στατιστικά χαρακτηριστικά. Καθώς αλλάζει και εξελίσσεται η συμπεριφορά των χρηστών στο

διαδίκτυο και παράλληλα και οι κακόβουλες επιθέσεις μαζί τους, τα σετ δεδομένων για την αξιόλογηση των IDS για να είναι αποτελεσματικά, πρέπει να γίνουν περισσότερο τροποποιήσιμα, επεκτάσιμα και αναπαράξιμα.[15]

Από τα παραπάνω γίνεται φανερό ότι τα IDS και τα αντίστοιχα σετ δεδομένων πρώτον χρειάζονται περαιτέρω βελτίωση, δεύτερον ιδιαίτερα σημαντική θα είναι και η ανάπτυξη νέων μεθόδων αντιμετώπισης των κυβερνοεπιθέσεων και τρίτον, που είναι ίσως και το πιο σημαντικό, είναι τα σετ δεδομένων να είναι διαθέσιμα για ευρεία χρήση, κάτι που θα επιτρέψει την σύγκριση των δυνατοτήτων των IDS, ως προς την αποτελεσματικότητα τους. [16] [17]

2.2.2.5 Διαθέσιμα σετ δεδομένων

Παρακάτω παρατίθενται τα πιο σημαντικά ήδη διαθέσιμα σετ δεδομένων από το 1998 με μια σύντομη περιγραφή των χαρακτηριστικών τους. Ακολουθούν τρεις πίνακες (Πίνακες 2.2, 2.3 και 2.4) που επιχειρούν συνοπτική σύγκριση των χαρακτηριστικών αυτών.

i. DARPA (Lincoln Laboratory 1998, 1999)

Η πρώτη προσπάθεια δημιουργίας σετ δεδομένων έγινε στο MIT από το Lincoln Lab και είναι γνωστά ως DARPA σετ δεδομένων (DARPA datasets): DARPA ‘98 και DARPA ‘99. Το DARPA ‘98 αποτελείται από 4 GB συμπιεσμένων tcpdump δεδομένων, που περιλαμβάνουν δικτυακή κίνηση 7 εβδομάδων και ίχνη 5 εκατομμυρίων συνδέσεων. Οι δύο πρώτες εβδομάδες προέρχονται από δοκιμαστική (test) κίνηση και περιλαμβάνουν 2 εκατομμύρια συνδέσεις. Το DARPA ‘99 αποτελείται από δικτυακή κίνηση 5 εβδομάδων καταγεγραμμένη από δύο διακριτά σημεία, ένα εντός του δικτύου και ένα εκτός. [18] Ερευνητές άσκησαν κριτική στα DARPA σετ δεδομένων ως προς, πρώτον την εκ των υστέρων εισαχθείσα και τεχνητή κακόβουλη κίνηση και δεύτερον ως προς τα χαρακτηριστικά της καλόβουλης κίνησης. Η κίνηση προήλθε από δραστηριότητες όπως η αποστολή και ανάγνωση μηνυμάτων ηλεκτρονικού ταχυδρομείου (email), την περιήγηση στο διαδίκτυο μέσω browser, την αποστολή και λήψη αρχείων μέσω FTP, τη χρήση telnet για την σύνδεση και εκτέλεση εργασιών σε απομακρυσμένους υπολογιστές, την αποστολή και λήψη IRC μηνυμάτων και την παρακολούθηση και καταγραφή μηνυμάτων SNMP. Η κακόβουλη κίνηση συμπεριλάμβανε DOS επιθέσεις, επιθέσεις τύπου guess password, buffer overflow, syn flood, Nmap κ.α. Συνολικά δεν αντιπροσωπεύει ωστόσο ρεαλιστική κίνηση δικτύων με χαρακτηριστική για παράδειγμα την απουσία ψευδώς θετικών συναγερμών (false positives) με αποτέλεσμα πλέον τα DARPA σετ δεδομένων να θεωρούνται παρωχημένα για την εκπαίδευση σύγχρονων IDS. [19] [20]

ii. KDD ‘99 (University of California, Irvine 1998-1999)

Το KDD Cup 1999 σετ δεδομένων δημιουργήθηκε από την επεξεργασία των tcmpdump δεδομένων του DARPA ‘98, και αποτελεί ουσιαστικά μια βελτιωμένη εκδοχή του. Συμπεριλαμβάνει νέες κακόβουλες επιθέσεις όπως Neptune-DoS, pod-DoS, Smurf-DoS και buffer overflow.[21] Οι καλόβουλες και κακόβουλες κινήσεις συγχωνεύτηκαν σε ένα περιβάλλον προσομοίωσης με αποτέλεσμα το σετ δεδομένων να περιέχει σε σημαντικό βαθμό περιττή ή παραπανίσια κίνηση με αλλοιωμένα

δεδομένα. Ως φυσικό συνεπακόλουθο τα αποτελέσματα του συγκεκριμένου σετ δεδομένων δεν ήταν τα επιθυμητά. Αργότερα έγινε μια προσπάθεια να καταπολεμηθούν τα παραπάνω προβλήματα με τον διάδοχο του KDD’ 99, το NSL-KDD που είχε σαφώς καλύτερα αποτελέσματα.[19] [22]

iii. DEFCON (The Shmoo Group, 2000)

Αρχικά δημιουργήθηκε το 2000 η πρώτη εκδοχή του, το DEFCON-8 σετ δεδομένων, το οποίο περιέχει επιθέσεις τύπου port scanning και buffer overflow. Στην δεύτερη εκδοχή του που δημιουργήθηκε το 2002 (DEFCON-10) προστέθηκαν επιπλέον επιθέσεις τύπου port scan, bad packets, FTP μέσω του telnet πρωτόκολλου κ.α. Η κίνηση που παρήχθει διαφέρει από την πραγματική κίνηση στο γεγονός ότι προέρχεται κατά κύριο λόγο από κακόβουλες επιθέσεις και όχι από καλόβουλη στο σύνολο κίνηση με σποραδικές επιθέσεις. [23] [3]

iv. CAIDA (Center of Applied Internet Data Analysis, 2002 - 2016)

Το CAIDA σετ δεδομένων αποτελείται από τρία διαφορετικά σετ:

- Το CAIDA OC48 που περιέχει διαφορετικούς τύπους δεδομένων όπως καταγράφηκαν στο OC48 δίκτυο (εξ’ ου και το όνομα), στον Σαν Χοσέ.
- Το CAIDA DDOS που περιέχει κίνηση μίας ώρας, ενώ το δίκτυο βρίσκεται υπό DDOS επίθεση. Η κίνηση αποθηκεύτηκε ανά πεντάλεπτο σε αρχεία τύπου pcap.
- Το CAIDA Internet Traces 2016 που είναι απλή καταγραφή κίνησης δικτύου υψηλών ταχυτήτων.

Τα CAIDA σετ δεδομένων έχουν το μειονέκτημα ότι είτε έχουν κίνηση που παρήχθει υπό ιδιαίτερες συνθήκες και συγκεκριμένες επιθέσεις (CAIDA OC48 και CAIDA DDOS), είτε έχουν χάσει σημαντικό μέρος της πληροφορίας τους για λόγους προστασίας των προσωπικών δεδομένων (CAIDA Internet Traces 2016). [3] [23]

v. LBNL (Lawrence Berkeley National Laboratory and ICSI, 2004 - 2005)

Πρόκειται για ίχνη του εσωτερικού δικτύου της LBNL. Το συγκεκριμένο σετ δεδομένων αν και περιέχει πραγματική κίνηση, έχει επίσης χάσει σημαντικό μέρος της πληροφορίας του για λόγους προστασίας προσωπικών δεδομένων. [24] [3]

vi. CDX (United States Military Academy, 2009)

Το CDX σετ δεδομένων ανέδειξε την δυνατότητα παραγωγής σετ δεδομένων από στρατιωτικά δίκτυα. Περιέχει τόσο καλόβουλη κίνηση (Web, email, DNS lookups κ.α.), όσο και κακόβουλη κίνηση που προέρχεται από επιθέσεις μέσω εργαλείων όπως τα Nikto, Nessus και WebScarab. Μπορεί να χρησιμοποιηθεί με ιδιαίτερη επιτυχία για τον έλεγχο της σωστής λειτουργίας των κανόνων συναγερμού των IDS συστημάτων αλλά έχει ελλείψεις ως προς την διαφορετικότητα και ποικιλία της κίνησης, καθώς όπως προαναφέθηκε προέρχεται μόνο από στρατιωτικό δίκτυο.[25]

vii. Kyoto (Kyoto University, 2009)

Το Kyoto σετ δεδομένων δημιουργήθηκε με τη χρήση honeypots[7] , κάτι που είχε ως αποτέλεσμα να μην χρειάζεται επεξεργασία για την προστασία προσωπικών δεδομένων. Αυτό το χαρακτηριστικό του ωστόσο είναι και το βασικό του μειονέκτημα καθώς έχει μόνο κίνηση που παγιδευόταν από τα honeypots. Επίσης η κίνηση του, προέρχεται μόνο από DNS και email πρωτόκολλα με αποτέλεσμα να μην υπάρχουν ψευδώς θετικοί συναγερμοί και άρα δεν μπορεί να συσχετιστεί με πραγματική κίνηση.[3] [22]

viii. Twente (University of Twente, 2009)

Το Twente σετ δεδομένων συμπεριλαμβάνει υπηρεσίες όπως OpenSSH, Apache web server και proftp καθώς και κίνηση που καταγράφηκε σε honeypot. Ταυτόχρονα, υπάρχουν κινήσεις που παρήχθησαν από πρωτόκολλα auth/ident, ICMP, IRC κ.α. οι οποίες δεν μπορούν να χαρακτηριστούν αμιγώς καλόβουλες ή κακόβουλες. Η κίνηση προσομοιάζει μεν στην πραγματική, ωστόσο είναι εμφανής η έλλειψη του πλουραλισμού και της έντασης των κακόβουλων επιθέσεων.[23] [26]

ix. UMASS (University of Massachusetts, 2011)

Το UMASS σετ δεδομένων αποτελείται από δεδομένα που παρήχθησαν από ένα TCP download request σενάριο επίθεσης. Η πρωτοτυπία του έγκειται στο γεγονός ότι έχει και κίνηση από δίκτυο με ασύρματες συσκευές. Δεν είναι ιδιαίτερα χρήσιμο για έλεγχο απόδοσης IDS και IPS συστημάτων εξαιτίας της απουσίας διαφορετικότητας στη κίνηση και στις επιθέσεις.[27] [28] [29]

x. ISCX2012 (University of New Brunswick, 2012)

Το ISCX2012 αποτελεί τομή στα σετ δεδομένων καθώς εισήχθη για πρώτη φορά η έννοια των άλφα και βήτα προφίλ. Το βήτα προφίλ[8] αποτελεί την γεννήτρια καλόβουλης κίνησης[9], που παράγει ρεαλιστική κίνηση δικτύου με τον αντίστοιχο θόρυβο. Το άλφα προφίλ[10] επιχειρεί πολλαπλά σενάρια κακόβουλων επιθέσεων και παράγει το αντίστοιχο τμήμα της κίνησης του δικτύου. Πολύ σημαντική διαδικασία, που επίσης εισήχθη με το συγκεκριμένο σετ δεδομένων είναι η προεργασία της ανάλυσης πραγματικής κίνησης για την εξαγωγή στατιστικών, που με τη σειρά τους αξιοποιούνται για την δημιουργία των προφίλ. Πρωτόκολλα για τα οποία εξήχθησαν στατιστικά είναι μεταξύ άλλων τα HTTP, SMTP, SSH, IMAP, POP3 κ.α. Αποτέλεσμα των παραπάνω, είναι η παραγωγή κίνησης ιδιαίτερα ρεαλιστικής, χωρίς να έχει χαθεί πληροφόρια για λόγους προστασίας των προσωπικών δεδομένων όπως σε πολλά από τα προγενέστερα σετ δεδομένων. Καθώς το συγκεκριμένο σετ δημιουργήθηκε το 2012, απουσιάζουν αρκετά πρωτόκολλα που κυριαρχούν στα σημερινά δίκτυα, όπως το HTTPS[11]. Επίσης, στα μειονεκτήματα του είναι ότι η κατανομή των επιθέσεων δεν προέκυψε από στατιστική ανάλυση αληθινής κίνησης. [15]

xi. ADFA (University of New South Wales, 2013)

7 Παγίδες που έχουν σαν στόχο να ανιχνεύσουν ή να εξουδετερώσουν κάθε μη εξουσιοδοτημένη πρόσβαση σε δίκτυα υπολογιστών [2]

8 Στην αγγλική βιβλιογραφία b-profile

9 Στην αγγλική βιβλιογραφία benign traffic

10 Στην αγγλική βιβλιογραφία α-profile

11 70% της σημερινής κίνησης είναι HTTPS [3]

Για την παραγωγή του συγκεκριμένου σετ δεδομένων έγινε εγκατάσταση, στο συστήμα που θα παρήγαγε την κίνηση, μεταξύ άλλων και οι παρακάτω υπηρεσίες: Apache, MySQL και Tikiwiki, καθώς Database και FTP server. Οι κακόβουλες επιθέσεις περιλάμβαναν FTP και SSH password bruteforce, Meterpereter, C100 Webshell κ.α. Η συμπεριφορά κάποιων από των επιθέσεων δεν ήταν σαφώς διακριτή από την καλόβουλη κίνηση.[30] [31]

xii. UNSW – NB15 (2015)

Για το UNSW – NB15 σετ δεδομένων χρησιμοποίηθηκε ένα εργαλείο, το IXIA PerfectStorm, για την παραγωγή αυτοματοποιημένων επιθέσεων, σε μια σειρά από server. Συλλέχθηκαν tcpdump ίχνη κίνησης δικτύου, για συνολικά 31 ώρες, που αντιστοιχεί σε περίπου 2 εκατομμύρια flows. Το βασικό πρόβλημα του συγκεκριμένου σετ δεδομένων είναι η συνθετική παραγωγή κίνησης από θεωρητικά μοντέλα, που απέχει αρκετά μία ρεαλιστική κίνηση. [32]

xiii. CSE – CIC – IDS (2018)

Το συγκεκριμένο σετ δεδομένων καλύπτει τρεις τύπους κακόβουλων επιθέσεων σε δίκτυα: Botnet, brute-force, Denial of Service (DoS), DdoS, infiltration και web attacks. Παρήχθει με βάση συνθετικά προφίλ χρηστών. Για την πραγματοποίηση των επιθέσεων χρησιμοποιήθηκαν 50 δικτυακοί κόμβοι, ενώ η τοπολογία του «θύματος» συμπεριλάμβανε 420 προσωπικούς υπολογιστές και 30 servers. [16]

Data-Set	Reference Datasets		More-Recent Datasets			
	DARPA'98 [70]	KDD Cup'99 [71]	DDoS 2016 [43]	UNSW-NB15 [59]	CICIDS 2017 [60]	UGR'16 [61]
Year	1998	1999	2016	2018	2017	2016
Type of traffic	Synthetic network traffic	Synthetic network traffic	Randomized to obtain realistic results	Synthetic network traffic	B-Profile system	Real network traffic with realistic attacks
Raw binary Data	4 GB	n/a	n/a	100 GB of the raw traffic	n/a	14 GB
How it was col-lected	tcpdump	built from the DARPA'98 dataset	A network simulator (NS2)	tcpdump and IXIA traffic generator PerfectStorm	user behavior based on FTP, email, HTTP, HTTPS, and SSH protocols	netflow traces
Col-lection time	7 weeks	n/a	n/a	31 h	5 days	more than 4 months
No. of records	5 M records	4.9 M single connection vectors	734,627 records	2 Million	n/a	16,900 M
Label-ed attack types	n/a	DoS, User to Root (U2R), Remote to Local (R2L) and Probing Attack	DDoS attack (HTTP Flood, SIDDOS, UDP Flood, and Smurf)	Fuzzers, Backdoors, DoS, Exploits, Generic, ReconnaissAnce, Shellcode, Worms	Botnet, Brute Force SSH, DoS, DDoS, FTP, Infiltration, Heartbleed, Web Attack	DoS, Scan, Botnet (synthetic), IP in blacklist, UDP Scan, SSH Scan, SPAM, anomaly

Πίνακας 2.2 Συνοπτική σύγκριση των χαρακτηριστικών διαφόρων σετ δεδομένων[16]

Network Attack	Datasets					
	DDoS 2016 [43]	UNSW-NB15 [59]	CICIDS 2017 [60]	UGR'16 [61]	NSL-KDD [38]	CSE-CIC-IDS2018 [62]
Fuzzers	No	Yes	No	No	No	No
Generic	No	Yes	Yes	No	No	Yes
Virus	No	No	No	No	No	No
Worm	No	Yes	No	No	No	No
Trojan	No	Yes	No	No	No	No
DoS	Yes	Yes	Yes	Yes	Yes	Yes
DDoS	Yes	No	Yes	No	No	Yes
Network Attack	No	No	No	No	No	No
Physical Attack	No	No	No	No	No	No
Information Gathering Attack	No	Yes	No	Yes	No	No
User to Root (U2R)	No	Yes	No	No	Yes	No
Remote to Local (R2L)	No	No	No	No	Yes	No
Probe	Yes	No	No	Yes	Yes	No
Brute-force	No	No	Yes	No	No	Yes
Web	No	No	Yes	No	No	Yes
Infiltration	No	Yes	Yes	No	No	Yes
Botnet	No	No	Yes	Yes	No	Yes

Πίνακας 2.3 Παρουσίαση των επιθέσεων που συμπεριλαμβάνουν σημαντικά σετ δεδομένων [16]

		DARPA	KDD'99	DEFCON	KAIDAs	LBNL	CDX	KYOTO	TWENTE	UMASS	ISCX2012	ADFA2013	Proposed Model
Network		Y	Y	N	Y	Y	N	Y	Y	Y	Y	Y	Y
Traffic		N	N	N	Y	Y	N	N	Y	N	N	Y	Y
Label		Y	Y	N	N	N	N	Y	Y	Y	Y	Y	Y
Interaction		Y	Y	Y	N	N	Y	Y	Y	N	Y	Y	Y
Capture		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Protocols	http	Y	Y	Y	–	Y	Y	Y	Y	Y	Y	Y	Y
	https	N	N	N	–	N	N	Y	N	N	N	N	Y
	ssh	Y	Y	Y	–	Y	Y	Y	Y	–	Y	Y	Y
	ftp	Y	Y	N	–	N	Y	Y	N	N	Y	Y	Y
	email	Y	Y	N	–	N	Y	Y	N	N	Y	Y	Y
Attacks	Browser	Y	Y	N	N	–	N	Y	N	N	Y	Y	Y
	Bruteforce	Y	Y	N	N	–	N	Y	Y	N	Y	Y	Y
	DoS	Y	Y	–	Y	–	Y	Y	N	–	Y	–	Y
	Scan	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	Backdoor	N	N	Y	N	–	N	Y	N	N	N	N	Y
	DNS	N	N	N	Y	–	Y	Y	N	N	N	N	Y
	Others	Y	Y	Y	Y	–	–	Y	Y	Y	Y	Y	Y
Anonymity		N	N	–	Y	Y	–	N	–	–	N	–	Y
Hetrogenity		N	N	N	N	N	N	N	–	–	Y	–	Y
Feature Set		N	Y	N	N	N	N	Y	N	N	N	N	Y
Metadata		Y	Y	N	Y	N	N	Y	Y	N	Y	Y	Y

Πίνακας 2.4 Συνοπτική σύγκριση των χαρακτηριστικών διαφόρων σετ δεδομένων [3]

2.2.2.6 Απαιτήσεις ενός σύγχρονου σετ δεδομένων

Οι απαιτήσεις ενός σύγχρονου σετ δεδομένων θα μπορούσαν να συνοψιστούν στα παρακάτω χαρακτηριστικά:

i. Λειτουργίες – Χαρακτηριστικά

   Τα σετ δεδομένων μπορούν να συμπεριλαμβάνουν τόσο χαρακτηριστικά δικτύου όπως διάρκεια των flows, χρονοσφραγίδες, πλήθος bytes και πακέτων, σημαίες, IP διευθύνσεις και αριθμό port κ.α., αλλά και χαρακτηριστικά host όπως πλήθος αποτυχημένων προσπαθειών σύνδεσης σε μια υπηρεσία, αντίστοιχα πλήθος επιτυχημένων προσπαθειών σύνδεσης, πλήθος σημαίων που υποδεικνύουν κάποιο πρόβλημα κ.α. Σημαντικό για να συμπεριλαμβάνεται ικανοποιητικό πλήθος των παραπάνω χαρακτηριστικών σε ένα σετ δεδομένων, είναι η πληροφορία να παρέχεται σε ακατέργαστη μορφή, όπως έχει καταγραφεί από τους sniffers του δικτύου. Σε πολλά από τα ήδη υπάρχοντα σετ δεν ικανοποείται αυτή η συνθήκη, καθώς όπως αναφέρθηκε ήδη πολλάκις, η πληροφορία αλλοιώνεται για λόγους προστασίας των προσωπικών δεδομένων.

ii. Πραγματική καλόβουλη κίνηση

   Είναι ιδιαίτερα σημαντικό το σετ δεδομένων να περιέχει πραγματική κίνηση, κυρίως όσον αφορά το τμήμα της καλόβουλης κίνησης. Όπως αναλύθηκε και παραπάνω, τα μαθηματικά μοντέλα φάνηκε να αστοχούν στις προσεγγίσεις τους με αποτέλεσμα η συνθετική κίνηση να ακολουθεί μοτίβα και να έχει κανονικοποιήμενη συμπεριφορά, που δεν προσιδιάζει σε κίνηση αληθινού δικτύου.

iii. Σύγχρονες κακόβουλες επιθέσεις

   Τα σετ δεδομένων που χρησιμοποιούνται για την εκπαίδευση IDS συστημάτων πρέπει να περιέχουν ρεαλιστικά σενάρια κακόβουλων επιθέσεων και κυρίως μοντέρνες και σύγχρονες επιθέσεις που ανταποκρίνονται στο σημερινό σύνθετο περιβάλλον. Η εκπαίδευση ενός IDS με σετ δεδομένων με παρωχημένες επιθέσεις μπορεί να αποβεί καταστροφική.

iv. Labelling

   Ιδιαίτερα σημαντικό είναι η κίνηση να έχει ετικετοποιηθεί σε καλόβουλη και κακόβουλη. Στην περίπτωση της τελευταίας η κίνηση πρέπει να είναι ετικετοποιημένη και με τον τύπο της επίθεσης. Η παραπάνω διαδικάσια στην πράξη αποδεικνύεται ιδιαίτερα δύσκολη.

v. Διάρκεια

   Ιδανικά τα σετ δεδομένων πρέπει να προέρχονται από καταγραφή που εκτείνεται τουλάχιστον σε μήκος εβδομάδας για να μπορούν να παρατηρηθούν αναμενόμενες διαφοροποιήσεις στην κίνηση ανάμεσα σε αυτή που καταγράφηκε μέρα και νύχτα, ανάμεσα σε καθημερινές - εργάσιμες μέρες και σε σαββατοκύριακα, μέρες με υψηλό φόρτο εργασίας και αντίστοιχα με χαμηλότερο κ.α.

vi. Documentation

Το σετ δεδομένων πρέπει να συνοδεύεται απαραίτητα από την αντίστοιχη τεκμηρίωση του (documentation), ώστε να μπορούν να παρατηρηθούν εξαρχής τυχών αδυναμίες και ελλείψεις αλλά και δυνατότητες περαιτέρω επέκτασης του σετ.

vii. Format

Οι πληροφορία στα σετ δεδομένων συνήθως εμπεριέχεται σε μορφή αρχείων (format) pcap , csv ή flow (netflow). Τα pcap αρχεία επιτρέπουν μια καλύτερη εκτίμηση για ένα σύστημα IDS καθώς περιέχουν ολόκληρη την καταγραφείσα πληροφορία. Στα csv αρχεία αντίθετα έχει συνήθως προηγηθεί επεξεργασία, μια διαδικασία που έχει καταστεί πολύ απλή πλέον μέσω βιβλιοθηκών όπως pandas στην γλώσσα python, κάτι που ενίοτε μπορεί και να χαρακτηριστεί και ως πλεονέκτημα αν η αφαίρεση πληροφορίας κρίνεται απαραίτητη, όπως για παράδειγμα για λόγους προστασίας προσωπικών δεδομένων. Σημαντικό είναι να σημειωθεί ότι η μορφή αρχείου που θα επιλεχθεί σχετίζεται άμεσα και με το μέγεθος των δεδομένων. [3]

2.6 Η έννοια του προφίλ

Από την πρώτη απόπειρα παραγωγής σετ δεδομένων (DARPA 1998 [18]) μέχρι και τις πιο σύγχρονες προσεγγίσεις, ως γεγονός – τομή, μπορεί να χαρακτηριστεί η εισαγωγή της έννοιας του προφίλ το 2012 από το University of New Brunswick με το σετ δεδομένων ISCX2012. [15]

Ένα προφίλ αποτελεί μια αφηρημένη αναπαράσταση διάφορων λειτουργιών και γεγονότων, με σκοπό την αναπαραγωγή συγκεκριμένων ρεαλιστικών συμπεριφορών, όπως αυτές καταγράφονται από το δίκτυο. Η αφηρημένη τους έννοια, τους επιτρέπει να είναι ανεξάρτητα από τοπολογίες και τους επιτρέπει να επαναχρησιμοποιούνται για την αναπαραγωγή δικτυακής κίνησης σε ένα μεγάλο εύρος διαφορετικών τοπολογιών και δικτύων. Ταυτόχρονα τους δίνει την δυνατότητα να είναι παραμετροποιήσιμα και να μπορούν να επιδείξουν πλουραλισμό στην δικτυακή συμπεριφορά τους. [15]

Τα προφίλ μπορούν να χρησιμοποιηθούν συνδιαστικά για να επιτύχουν την εκάστοτε επιθυμητή συμπεριφορά. Αρκετές κατηγοριοποιήσεις τους έχουν επιχειρηθεί από την ακαδημαϊκή κοινότητα, με μία όμως να είναι η κυρίαρχη: ο διαχωρισμός τους σε άλφα και βήτα.

2.6.1 α-Προφίλ

Τα α-προφίλ (a-profiles) περιγράφουν σενάρια κακόβουλης επιθετικής κίνησης. Στις απλούστερες περιπτώσεις τα σενάρια υλοποιούνται χειρωνακτικά από χρήστες πάνω από την καλόβουλη κίνηση ενός δικτύου. Σε άλλες περιπτώσεις οι επιθέσεις γίνονται από αυτοματοποιημένα συστήματα, αλλά συνήθως με λιγότερη επιτυχία, καθώς η δικτυακή κίνηση σε αυτές τις περιπτώσεις εμφανίζει αποκλίσεις από την πραγματική. [15] Τα α-προφίλ δεν θα αποτελέσουν περαιτέρω αντικείμενο αυτής της εργασίας και αναφέρθηκαν για λόγους πληρότητας.

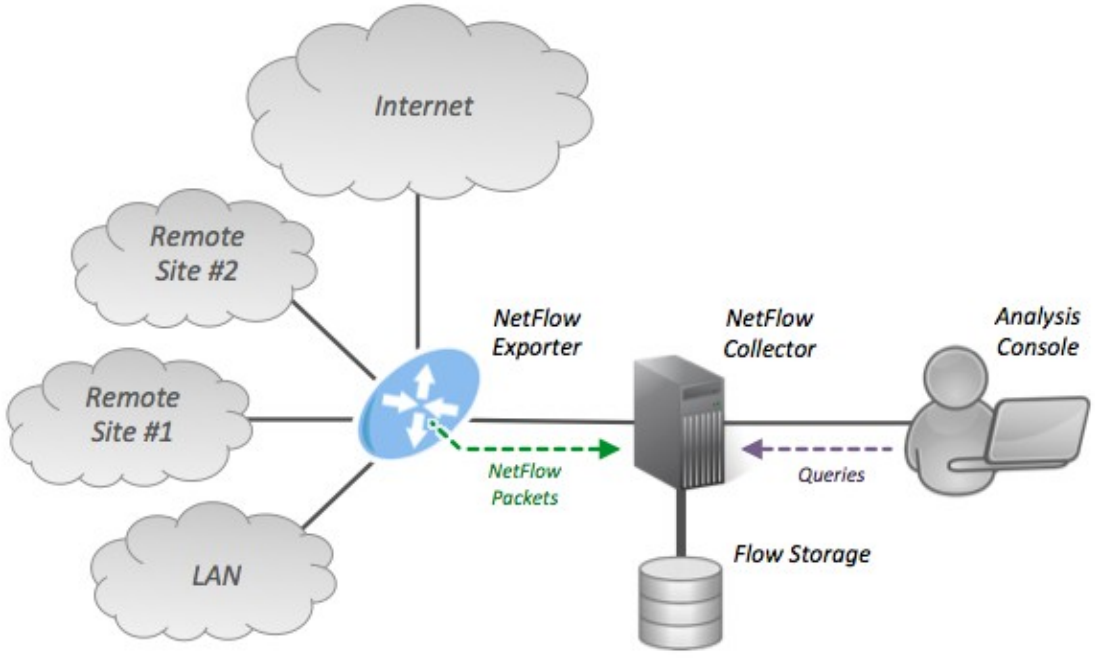
2.6.2 β-Προφίλ

Τα β-προφίλ (b-profiles) περιγράφουν σενάρια καλόβουλης μη επιθετικής κίνησης. Σκοπός τους είναι η παραγωγή ρεαλιστικής δικτυακής κίνησης που θα αποτελέσει το πλαίσιο ή υπόβαθρο για την υλοποίηση των κακόβουλων επιθέσεων που θα παράξουν τα α-προφίλ.
Για την αναπαραγωγή του έχουν χρησιμοποιηθεί μαθηματικές κατανομές ή και στατιστικές αναλύσεις πραγματικής κίνησης. Σε αντίθεση με τα α-προφίλ για την επιτυχημένη αναπαραγωγή τους δεν είναι υποχρεωτική η παρέμβαση του ανθρώπινου παράγοντα, πέραν και έπειτα του προαπαιτούμενου στάδιου του καθορισμού του τρόπου παραγωγής της δικτυακής κίνησης. [23]

Στις αναλύσεις πραγματικής δικτυακής κίνησης που ήδη έχουν γίνει από ερευνητές και ινστιτούτα έχει παρατηρηθεί ότι τα πρωτόκολλα που κυριαρχούν είναι το HTTP και HTTPS. [15] Πέραν αυτών, άλλα σηματικά πρωτόκολλα που παρατηρήθηκαν είναι τα: SMTP, IPOP, IMAP, SSH και FTP. [3]

2.7 Η έννοια του flow

Για την αναπαραγωγή ρεαλιστικής κίνησης απο τα β-προφίλ, αξιοποιούνται παρόμοιοι μηχανισμοί με αυτούς των web-crawlers[12]. Η κίνηση αυτή αποθηκεύεται σε αρχεία με format pcap, csv ή netflow. [18]



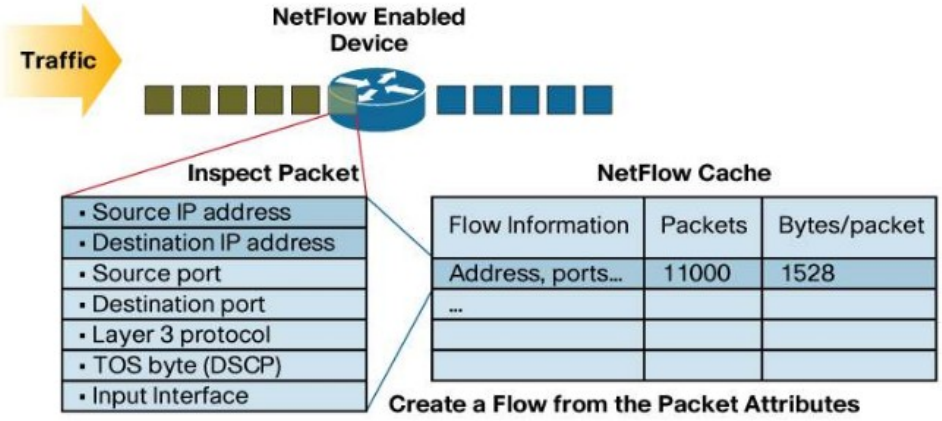
Εικόνα 2.2 Καταγραφή δικτυακής κίνησης σε format netflow[13]

12 Στα Ελληνική βιβλιογραφία αναφέρεται ως “ανιχνευτής ιστού” ή “αράχνη”. Είναι ένα πρόγραμμα που κυκλοφορεί στο διαδίκτυο με απώτερο σκοπό την συλλογή στοιχείων[1]

13 [Πηγή: https://en.wikipedia.org/wiki/NetFlow]

Τα netflow αρχεία εισήχθηκαν από την cisco το 1996 και δίνουν την δυνατότητα καταγραφής IP δικτυακής κίνησης καθώς αυτή εισέρχεται ή εξέρχεται από μια διεπαφή. Περιέχουν flows, δηλαδή μια ακολουθία από πακέτα που έχουν τα εξής κοινά χαρακτηριστικά:

i. Layer 3 τύπος πρωτοκόλλου
ii. IP source διεύθυνση
iii. IP destination διεύθυνση
iv. Ρούτερ ή switch διεπαφή
v. Source Port για UDP ή TCP
vi. Destination Port για UDP ή TCP
vii. IP Type of Services [33]



Εικόνα 2.3 Δημιουργία ενός flow[14]

2.8 APIs

Μια διεπαφή προγραμματισμού υπολογιστών ή αλλιώς API[15], είναι ένα σύνολο από διασυνδέσεις μεταξύ εφαρμογών, συστημάτων και συσκευών. Αποτελεί ένα πλαίσιο επικοινωνίας μεταξύ τους, αυστηρά καθορισμένο ως προς τη μορφή των μηνυμάτων που επιτρέπεται να ανταλλάσσονται, το περιεχόμενο που μπορεί αυτά να έχουν, και τον τρόπο που δημιουργούνται και τελικά διακινούνται. Στην πράξη είναι τμήματα λογισμικού ή υλικού που χρησιμοποιούνται με σκοπό την απλούστευση της διαδικασίας προγραμματισμού, μέσω της αφαιρετικής παρουσίασης των λειτουργιών ενός συστήματος και της εύκολης αξιοποίησής τους από τους χρήστες [34].

2.8.1 RESTful APIs

Το πιο δημοφιλές αρχιτεκτονικό στυλ διεπαφών προγραμματισμού υπολογιστών είναι το REST (Representational State Transfer), που ορίστηκε το 2000 από τον Fielding [35] ως ένα σύνολο περιορισμών και κατευθυντήριων γραμμών σχεδιασμού. Πιο συγκεκριμένα, για

14 [Πηγή: http://www.service-desk.co/white_papers/cisco_netflow.pdf]
15 Application Programming Interface

να χαρακτηριστεί ένα σύστημα RESTful, δηλαδή να είναι τύπου REST, οφείλει να πληροί τους εξής περιορισμούς:

i. Αρχιτεκτονική Πελάτη-Διακομιστή (Client-Server)
Διαχωρίζοντας το περιβάλλον του χρήστη από αυτό των δεδομένων, βελτιώνουμε τόσο την φορητότητα του περιβάλλοντος του χρήστη μεταξύ πλατφορμών, όσο και την επεκτασιμότητα, κάτι που αποτελεί ισχυρό πλεονέκτημα για τις απαιτήσεις του Διαδικτύου.

ii. Έλλειψη Κατάστασης (Statelessness)
Κάθε αίτημα από τον πελάτη προς τον διακομιστή οφείλει να περιλαμβάνει όλες τις απαραίτητες πληροφορίες για την εκτέλεση των διεργασιών. Έτσι επιτυγχάνεται ευκολότερη παρακολούθηση των εντολών, καλύτερη αξιοπιστία σε περιπτώσεις ανάκτησης δεδομένων και βελτιωμένη επεκτασιμότητα, αφού απλοποιείται η υλοποίηση των επιμέρους συστημάτων. Ο συμβιβασμός για όλα αυτά είναι η επαναλαμβανόμενη αποστολή ορισμένων δεδομένων από την πλευρά του χρήστη.

iii. Δυνατότητα αξιοποίησης κρυφής μνήμης (Cacheability)
Η προσωρινή αποθήκευση δεδομένων στην κρυφή μνήμη (cache) από τον χρήστη και τα ενδιάμεσα συστήματα βελτιώνει την απόδοση του δικτύου, αποφεύγοντας την περιττή λήψη δεδομένων, όμως εγκυμονεί κινδύνους κατοχής προηγούμενων εκδόσεων αρχείων, σε περίπτωση που αυτά έχουν ενημερωθεί από τον διακομιστή αλλά δεν έχουν ληφθεί ανανεωμένα από τον χρήστη.

iv. Ομοιόμορφη Διεπαφή (Uniform Interface)
Οι ομοιόμορφες διεπαφές μεταξύ των τμημάτων του συστήματος προσφέρουν απλούστερη αρχιτεκτονική και καλύτερη εποπτεία των αλληλεπιδράσεων. Το αρνητικό όμως είναι η μείωση της αποδοτικότητας, μιας και η πληροφορία μεταδίδεται τυποποιημένη,αγνοώντας τις ιδιαιτερότητες κάθε εφαρμογής.

v. Πολυεπίπεδο Σύστημα (Layered System)
Μεταξύ πελάτη και διακομιστή μπορεί να παρεμβληθεί οποιοδήποτε ενδιάμεσο στάδιο, χωρίς να υπάρχει εμφανής επίδραση στην επικοινωνία και χωρίς να απαιτούνται τροποποιήσεις στις υλοποιήσεις των τελικών συστημάτων.

vi. Κώδικας κατά παραγγελία (Code-On-Demand)
Ο τελευταίος περιορισμός είναι ο μόνος προαιρετικός και αφορά στη δυνατότητα του χρήστη να κατεβάσει και να εκτελέσει τμήματα κώδικα με τη μορφή μικροεφαρμογών.

Τα RESTful Web APIs είναι προγραμματιστικές διεπαφές που ικανοποιούν τα παραπάνω κριτήρια και λειτουργούν στον Παγκόσμιο Ιστό [36]. Συνεπώς οι διεπαφές αυτού του είδους έχουν τα εξής χαρακτηριστικά [37]:

i. Εχουν μία διεύθυνση βάσης (base URL), όπως https://www.example.com/api
ii. Χρησιμοποιούν τις μεθόδους HTTP, όπως αναφέρονται παρακάτω [38] [39] [40] [41]:

- GET
  Τα GET requests χρησιμοποιούνται για την απεικόνιση της πληροφορίας του server.

- POST
  Με ένα POST request ο client έχει τη δυνατότητα να στείλει πληροφορία στον server. Ο server διαβάζει το request body, ελέγχει αν τηρεί τις προδιαγραφές (δομή, τύπος δεδομένων κλπ), εκτελεί τον κώδικα της κρίσιμης λογικής (business logic) και απαντάει στο χρήστη με κάποιο response body.

- PUT
  Για την τροποποίηση ήδη υπάρχοντος πληροφορίας (Update/Replace).

- DELETE
  Για την διαγραφή πληροφορίας απο τον server.

- PATCH
  Για την μερική τροποποίηση μιας εγγραφής (partial update/Modify).

  Οι απαντήσεις του server συνοδεύονται με συγκεκριμένο HTTP κωδικό κατάστασης (status code), ανάλογα με την λειτουργικότητα και την επιτυχία του κάθε request. Οι κωδικοί αυτοί χωρίζονται σε 5 κατηγορίες, όπως φαίνεται παρακάτω:

  • 1xx – Informational response
  • 2xx – Success
  • 3xx – Redirection
  • 4xx – Client Errors
  • 5xx – Server Errors

iii. Περιλαμβάνουν στα μηνύματά τους αναγνωριστικό για τη μορφή των δεδομένων που αποστέλλονται, όπως για παράδειγμα text/html ή application/json

Οι περισσότερες εταιρειές πληροφορικής σήμερα κάνουν χρήση προγραμματιστικών διεπαφών διαδικτύου τύπου REST, όπως για παράδειγμα οι Google[16], Twitter[17] και Wikipedia[18]. Η πιο συνηθισμένη χρήση είναι η δυνατότητα που προσφέρουν σε οποιονδήποτε προγραμματιστή επιθυμεί να αλληλεπιδράσει με τις εφαρμογές τους μέσω των διεπαφών τους. Σε αυτή την περίπτωση δημιουργούν μια σειρά από τελικά σημεία (endpoints), δηλαδή σημεία με τα οποία μπορεί κάποιος να αλληλεπιδράσει με τη διεπαφή. Αυτά έχουν τη μορφή ξεχωριστών διευθύνσεων που επεκτείνουν τη διεύθυνση βάσης, όπως για παράδειγμα https://www.example.com/api/users/, και προσφέρουν λειτουργίες που πραγματοποιούνται με την αποστολή και λήψη HTTP μηνυμάτων από και προς την ίδια διεύθυνση.[42]

16 [https://developers.google.com/gmail/api/reference/rest]
17 [https://developer.twitter.com/en/docs/api-reference-index]
18 [https://www.mediawiki.org/wiki/API:REST_API]

2.8.2 Swagger

Πρόκειται για μία γλώσσα περιγραφής διεπαφών (Interface Description Language) που περιγράφει RESTful APIs χρησιμοποιώντας το format JSON. Αναπτύχθηκε από την Wordnik το 2011 με σκοπό την ιδιωτική του χρήση από τον οργανισμό για την δημιουργία του developer.wordnik.com. Πρόκειται για ένα σύνολο κανόνων για την περιγραφή, παραγωγή, κατανάλωση και απεικόνιση RESTful υπηρεσιών που τελικά καθορίζει ένα πρότυπο διεπαφής ανεξάρτητο από την γλώσσα προγραμματισμού που έχει χρησιμοποιηθεί. Επιτρέπει τόσο σε ανθρώπους όσο και υπολογιστές να κατανοήσουν την λειτουργικότητα της υπηρεσίας που προσφέρει ο server χωρίς να έχουν πρόσβαση στον πηγαίο κώδικα, καθώς έχει σχεδιαστεί για να επιτελεί και ρόλο documentation. [43] [44]

Κεφάλαιο 3

Μεθοδολογία Μοντελοποίησης Κίνησης

3.1 Καταγραφή και παρατήρηση flows με το ntopng

Με δεδομένο ότι προσπαθούμε να επιτύχουμε την καλύτερη δυνατή προσέγγιση της ανθρώπινης συμπεριφοράς στο διαδίκτυο, έγινε αναπαραγωγη διαφόρων flows, καταγραφή τους μέσω του εργαλείου ntopng[19] και παρατήρηση των χαρακτηριστικών τους.

Τα συμπεράσματα που προέκυψαν από την παραπάνω ανάλυση είναι τα παρακάτω:

i. Με την έναρξη της επικοινωνίας του υπολογιστή μας με τον εκάστοτε server, καταγράφηκαν παραπάνω από ένα flows, εκ των οποίων τα περισσότερα ήταν βραχύβια (χρόνος ζωής της τάξεως των millisecond ή λίγων δευτερολέπτων). Για παράδειγμα για την αναπαραγωγή ενός βίντεο στο youtube δημιουργήθηκαν δύο TLS flows με διάρκεια όση και το βίντεο, όπου το μεν πρώτο ανιστοιχούσε στα streaming services και είχε και τον κύριο όγκο της κίνησης σε byte, το δε δεύτερο δεν μετέφερε κάποια πληροφορία, ενώ παράλληλα δημιουργήθηκαν και αρκετά άλλα, τα οποία ήταν μικρά σε διάρκεια και αντιστοιχούσαν σε διάφορα microservices όπως μεταφορά των πακέτων για τα thumbnails, διαφημίσεων κ.α.
ii. Για κάθε επικοινωνία με τον εκάστοτε server δημιουργούνται και αντίστοιχος αριθμός από flows. Για παράδειγμα ξεχωριστές αναζητήσεις στο google δημιουργούν τα δικά τους flows ακόμα και αν έγιναν ταυτόχρονα.
iii. Ένα flow σταματάει να καταγράφεται 15 δευτερόλεπτα από την τελευταία μεταφορά πακέτων μέσω ενός προκαθορίσμενου χρονοδιακόπτη (inactive flow timer). Επίσης διακόπτεται η καταγραφή ενός flow, τριάντα λεπτά μετά την εκκίνηση της καταγραφής του, ακόμα και αν αυτό είναι ενεργό, μέσω ενός άλλου προκαθορισμένου χρονοδιακόπτη (active flow timer). Τα παραπάνω αφενός περιγράφονται και στο αντίστοιχο documentation της cisco [33] , αφετέρου διαπιστώθηκαν και πειραματικά.
iv. Ανανέωση της ιστοσελίδας μέσω του browser ή κάποια ενέργεια του χρήστη που απαιτεί επικοινωνία με τον εκάστοτε server και άρα μεταφορά πακέτων, επανεκκινεί τον inactive flow timer.

3.2 Μοντελοποίηση της δικτυακής κίνησης ενός προφίλ

Για την ρεαλιστική αναπαραγωγή δικτυακής κίνησης, καθοριστικό ρόλο παίζει η διάρκεια των flows που θα αναπαράγουμε, καθώς και το χρονικό διάστημα που διαμεσολαβεί ανάμεσα στην επανέναρξη τους. Για την μοντελοποίηση των δύο παραπάνω παραμέτρων μπορούν να αξιοποιηθούν δύο λίστες (ή αντίστοιχα πίνακες): η duration list και η interarrivals list. Η μεν πρώτη περιέχει τις διάρκειες των εκάστοτε flow που θα αναπαραχθουν και η δεύτερη την διάρκεια διαλειμμάτων που διαμεσολαβεί ανάμεσα στο τερματισμό ενός flow και την εκκίνηση του επόμενου. Τα τελευταία πρέπει να είναι τουλάχιστον δεκαπέντε δευτερόλεπτα ώστε να έχει διασφαλιστεί ότι έχει τελείώσει η καταγραφή του τελευταίου flow. Για τις τιμές που μπορεί να λάβει η duration list παρουσιάζονται οι παρακάτω δύο προσεγγίσεις: χρήση ιστοσελίδων με στατιστικά πλοήγησης χρηστών στο διαδίκτυο και η στατιστική ανάλυση πραγματικής δικτυακής κίνησης.

19 https://www.ntop.org/products/traffic-analysis/ntop/

3.3 Ιστοσελίδες με στατιστικά για την πλοήγηση χρηστών στο διαδίκτυο

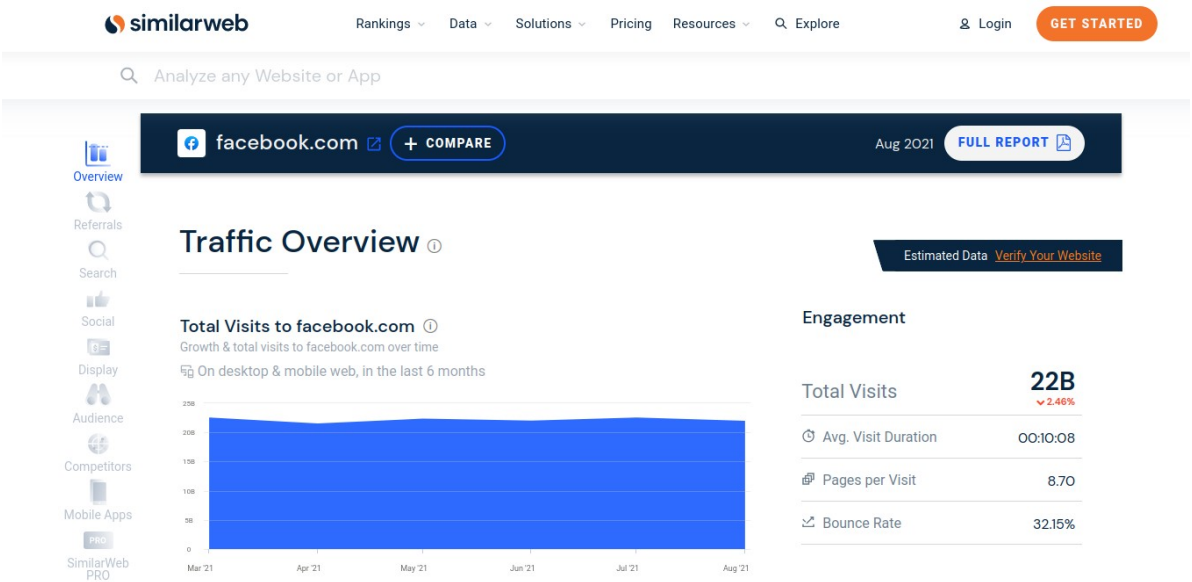
Ένας τρόπος για να λάβουμε χρόνους διάρκειας για τα flows είναι η χρήση ιστοσελίδων που καταγράφουν στατιστικά για την επισκεψιμότητα, την δικτυακή κίνηση και την πλοήγηση των χρηστών στο διαδίκτυο γενικότερα. Μεταξύ άλλων περιέχουν και τους εξής τρεις δείκτες για την επισκεψιμότητα μιας ιστοσελίδας:

i. Average Visit Duration, που είναι ο μέσος χρόνος παραμονής των χρηστών στον εκάστοτε ιστότοπο.
ii. Pages per Visit, που είναι πόσες διαφορετικές σελίδες του εκάστοτε ιστότοπου κατά μέσο όρο θα επισκεφτεί ένας χρήστης.
iii. Bounce Rate, που εκφράζει ως ποσοστό την πιθανότητα να μην συνεχίσει την πλοήγηση του ένας χρήστης στον εκάστοτε ιστότοπο πέραν της αρχικής σελίδας.

Οι τρεις αυτοί δείκτες μπορούν να αξιοποιηθούν για την μοντελοποίηση δικτυακής κίνησης ως εξής: ο πρώτος δείκτης (Average Visit Duration) να καθορίζει τον χρόνο διάρκειας ενός flow, ο δεύτερος (Pages Per Visit) να καθορίζει πόσες σελίδες ενός ιστότοπου θα επισκεφθούν στην συνολική διάρκεια του flow, και ο τρίτος να παρέχει την πιθανότητα η προσομοίωση να παραμείνει στην αρχική σελίδα του εκάστοτε ιστότοπου για όλη τη διάρκεια του flow.

Παραδείγματα τέτοιων ιστοσελίδων είναι:

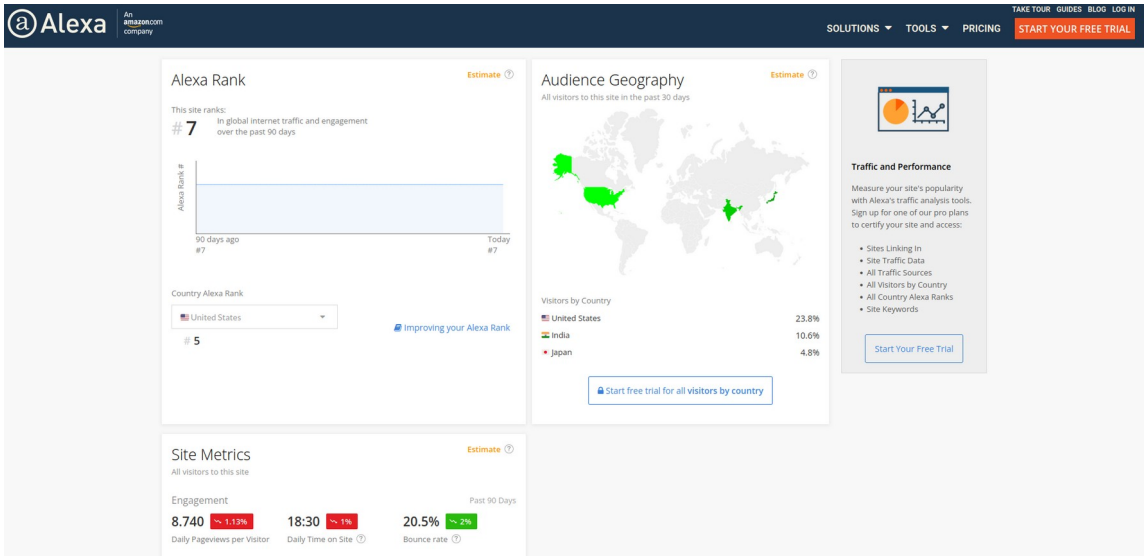
i. Similar Web (https://www.similarweb.com/)



Εικόνα 3.3 Ανάλυση δικτυακής κίνησης του similarweb.com για την ιστοσελίδα facebook.com[20]

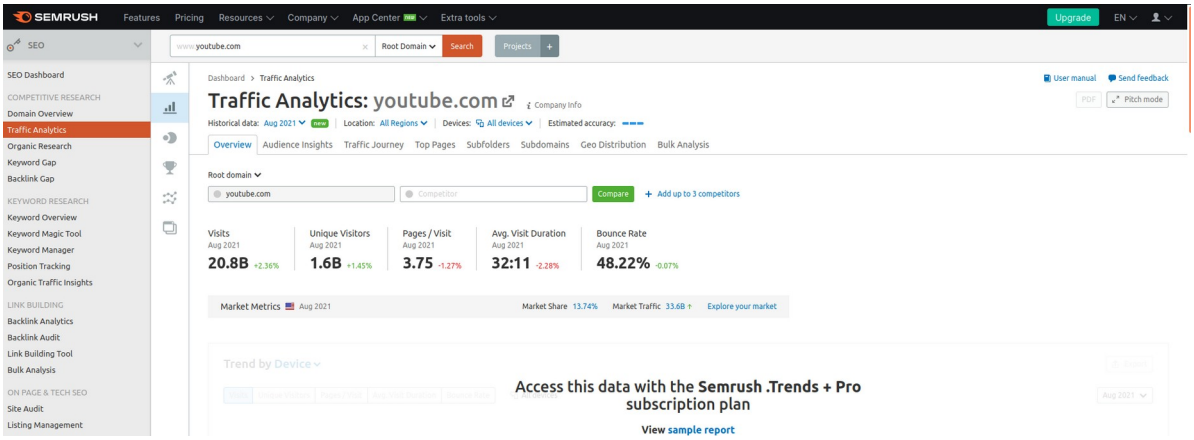
20 [Πηγή: https://www.similarweb.com/website/facebook.com/]

ii. Alexa (https://www.alexa.com/siteinfo)



Εικόνα 3.4 Ανάλυση δικτυακής κίνησης του alexa.com για την ιστοσελίδα youtube.com[21]

iii. Semrush (https://www.semrush.com/analytics/traffic/)



Εικόνα 3.5 Ανάλυση δικτυακής κίνησης του semrush.com για την ιστοσελίδα youtube.com[22]

Το βασικό πλεονέκτημα που παρέχουν αυτές οι ιστοσελίδες είναι ότι μπορεί να αυτοματοποιηθεί η διαδικασία της εξαγωγής της πληροφορίας, που κατόπιν θα

21 [Πηγή: https://www.alexa.com/siteinfo/youtube.com]
22 [Πηγή: hhttps://www.semrush.com/analytics/traffic/overview/youtube.com?searchType=domain]

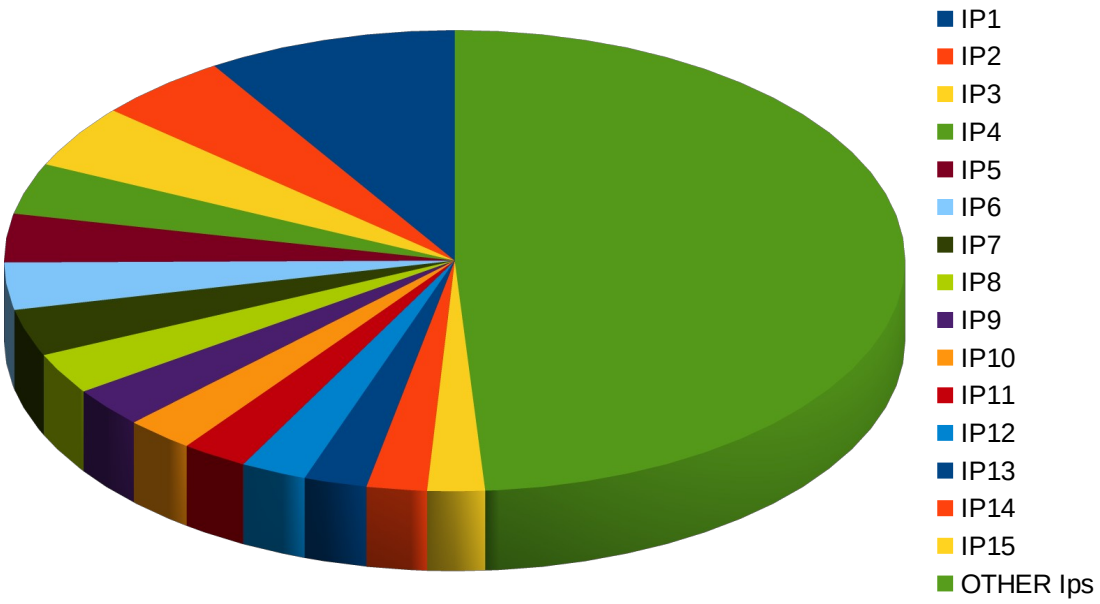
χρησιμοποιηθεί ως διάρκεια ενός flow. Οι περισσότερες μάλιστα παρέχουν και API, συνήθως κατόπιν συνδρομής, για την παραπάνω διαδικασία. Στον αντίποδα, το βασικό τους μειονέκτημα είναι ότι η πληροφορία που παρέχουν προέρχεται από στατιστικές αναλύσεις με βάση την χρήση τους παγκοσμίως από ένα ιδιαίτερα ετερόκλητο κοινό και μακριά από συγκεκριμένους ρόλους (όπως για παράδειγμα το χρόνο χρήσης του ίδιου ιστότοπου από το προσωπικό ενός νοσοκομείου, που εν προκειμένω μας ενδιαφέρει).

3.4 Στατιστική ανάλυση πραγματικής δικτυακής κίνησης

Μια δεύτερη προσέγγιση για να λάβουμε χρόνους διάρκειας για τα flows όπως και άλλα στατιστικά είναι η ανάλυση πραγματικής δικτυακής κίνησης νοσοκομείου. Στην παρούσα εργασία έγινε ανάλυση κίνησης που παρήχθει σε νοσοκομειακό συγκρότημα, όπου για λόγους προστασίας των προσωπικών δεδομένων δεν θα αναφέρουμε το όνομα του. Ο αλγόριθμος που ακολουθήθηκε, μαζί με παραδείγματα από την ανάλυση που έγινε, είναι ο παρακάτω:

i. Απομόνωση δικτυακής κίνησης που έχει σαν προορισμό κάποια IP, εφόσον γνωρίζουμε εξαρχής ότι επικοινωνία με την συγκεκριμένη διεύθυνση θα έχει προέλθει από μία οικογένεια προφίλ (πχ τα προφίλ που αντιστοιχούν στο φαρμακείο).

ii. Ανάλυση της παραπάνω δικτυακής κίνησης με βάση αυτή τη φορά τις source IP. Στην εικόνα 3.6 φαίνεται για παράδειγμα, κυκλικό διάγραμμα με τις source IP που είχαν επικοινωνία με IP που αντιστοιχεί με υπηρεσία φαρμακείου. Επίσης στον Πίνακα 3.1 φαίνονται τα στατιστικά που προήλθαν από την παραπάνω ανάλυση. Για λόγους προστασίας των προσωπικών δεδομένων οι πραγματικές IP έχουν αντικατασταθεί από ονόματα μεταβλητών IP1, IP2 κ.λ.π.



Εικόνα 3.6 Κυκλικό διάγραμμα με τις source IP που είχαν επικοινωνία με IP που αντιστοιχεί σε υπηρεσία φαρμακείου.

Source IP	**Συχνότητα εμφάνισης στην δικτυακή κίνηση**
IP1	87
IP2	46
IP3	43
IP4	35
IP5	33
IP6	32
IP7	32
IP8	28
IP9	27
IP10	24
IP11	23
IP12	23
IP13	22
IP14	21
IP15	20
Other IPs	475

Πίνακας 3.1 Πίνακας με την συχνότητα εμφάνισης των source IPs στην δικτυακή κίνηση

iii. Επιλογή των IP με την μεγαλύτερη συχνότητα εμφάνισης και εκ νέου απομόνωση της κίνησης που συμπεριλαμβάνει την εκάστοτε IP έιτε ως source, είτε ως destination. Κάθε δικτυακή κίνηση που απομονώνεται σε αυτό το βήμα χαρακτηρίζεται ως κίνηση ενός συγκεκριμένου προφίλ.
Για παράδειγμα με βάση τον πίνακα 3.1 έχουμε το προφίλ pharmacy1 με δικτυακή κίνηση που περιέχει ως source ή destination την IP1, το προφίλ pharmacy2 με δικτυακή κίνηση που περιέχει ως source ή destination την IP2, το προφίλ pharmacy3 με δικτυακή κίνηση που περιέχει ως source ή destination την IP3 κ.ο.κ.

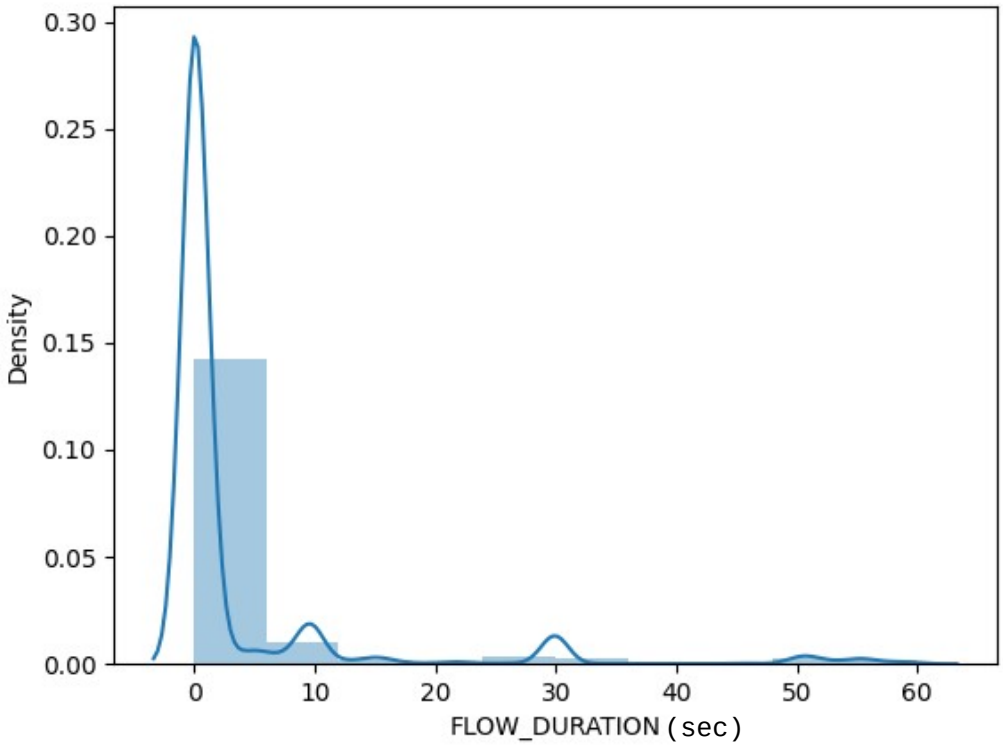
iv. Στατιστική ανάλυση της δικτυακής κίνησης των παραπάνω προφίλ. Εξαγωγή στατιστικών για:

(a) τη διάρκεια των flows
(b) την αιτία διακοπής των flows
(c) τα layer 7 πρωτόκολλα που παρατηρήθηκαν

(d) το πλήθος των bytes των flows που είχαν προέλευση από την συγκεκριμένη IP
(e) το πλήθος των bytes των flows που είχαν ως προορισμό την συγκεκριμένη IP
(f) το πλήθος των bytes των flows που είτε παρήχθησαν είτε είχαν ως προορισμό την συγκεκριμένη IP
(g) την βάρδια στην οποία καταγράφηκε το κάθε flow[23]

Παρατίθεται ενδεικτικά η ανάλυση για το προφίλ pharmacy1:

(a) Διάρκεια των flows



Εικόνα 2.8 Διάγραμμα Density – Flow Duration[24]

Τα στατιστικά που προέκυψαν είναι τα παρακάτω:

```
Mean: 3.61
Max: 59.93
Min: 0.00
Standard deviation: 10.28
Median: 0.07
```

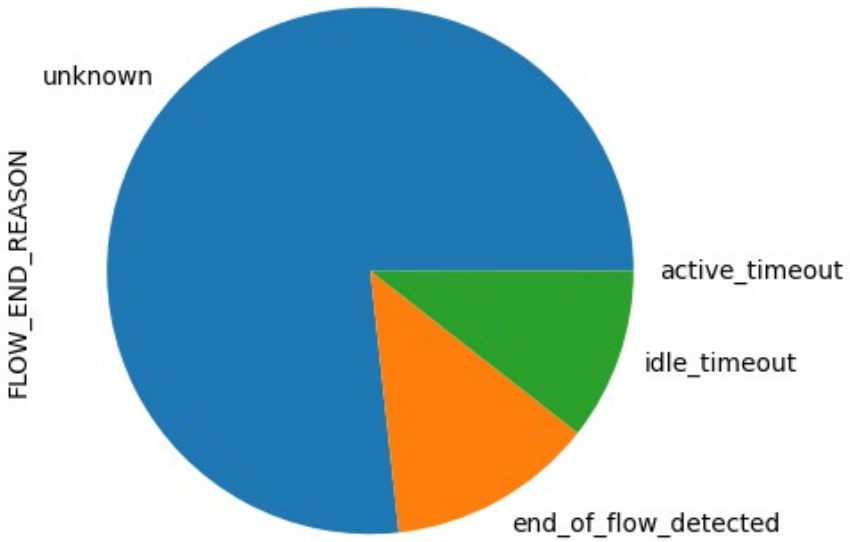
Παρατηρήθηκε μέσος χρόνος των flow είναι 3.61 δευτερόλεπτα, το μεγαλύτερο flow έχει διάρκεια 59.93 δευτερόλεπτα (στο ntopng είχε ρυθμιστεί ο active flow timer να είναι 60 seconds, σε αντίθεση με την προκαθορισμένη τιμή των 30 λεπτών),

23 Σαν σύμβαση για τις βάρδιες χρησιμοποιήθηκε η εξής: οι ώρες ενός 24ωρου, χωρίστηκαν σε τρεις οκτάωρες βάρδιες, 00:00-08:00 η πρώτη βάρδια, 08:01-16:00 η δεύτερη και 16:01-23:59 η τρίτη βάρδια.

24 Επιδεικνύεται και το kernel density estimation με σκοπό να φανεί η μεγάλη συγκέντρωση τιμών γύρω από το μηδέν. Οι αρνητικές τιμές πρέπει να αγνοηθούν από τον αναγνώστη.

το μικρότερο 0 δευτερόλεπτα, τυπική απόκλιση 10.28 δευτερόλεπτα και η διάμεσος είναι 0.07 δευτερόλεπτα.

(b) Αιτία διακοπής των flows



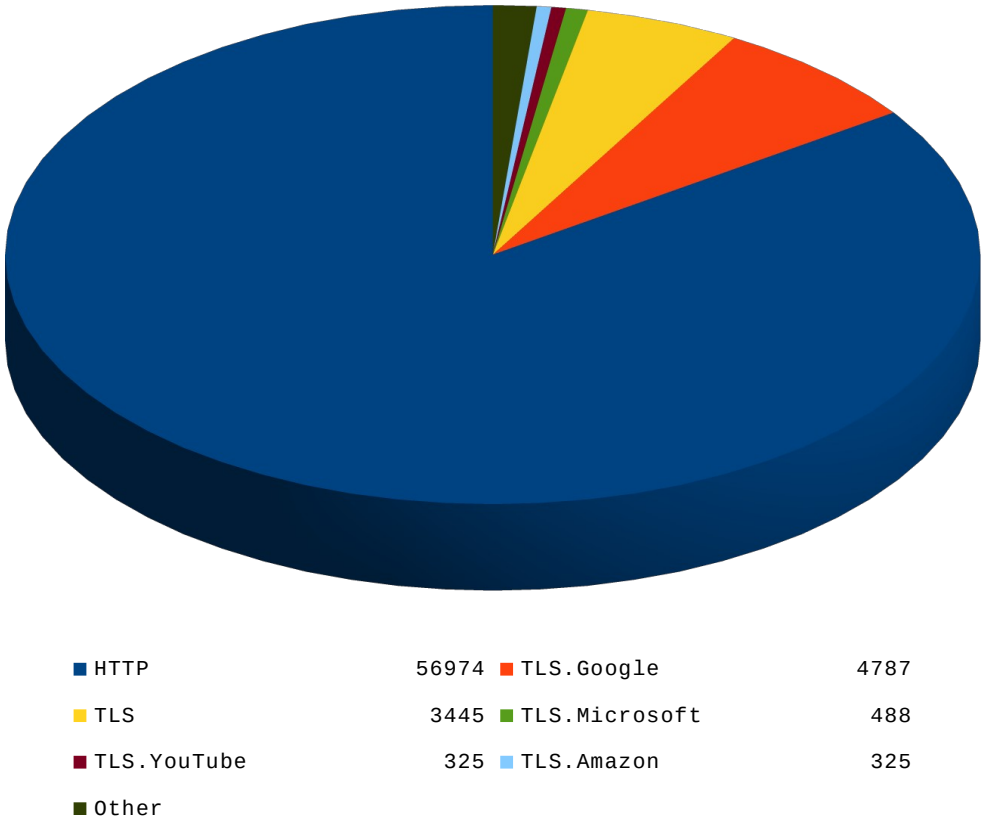
Εικόνα 2.9 Κυκλικό διάγραμμα αιτίας διακοπής των flows

Τα στατιστικά που προέκυψαν είναι τα παρακάτω:

```
unknown                 51646
end_of_flow_detected    8558
idle_timeout            7103
active_timeout          5
```

Παρατηρήθηκε ότι για τα flows στα οποία καταγράφηκε αιτία διακοπής, αυτή ήταν κατά κύριο λόγο η διακοπή του flow από τον χρήστη και διακοπή εξαιτίας του inactive flow timer. Αμελητέος ήταν ο αριθμός των flows η καταγραφή των οποίων διακόπηκε εξαιτίας του active flow timer.

(c) Layer 7 πρωτόκολλα που παρατηρήθηκαν



Εικόνα 2.10 Κυκλικό διάγραμμα των layer 7 πρωτόκολλων που παρατηρήθηκαν

Τα στατιστικά που προέκυψαν είναι τα παρακάτω:

L7 Protocol	**Πλήθος flows**
HTTP	56974
TLS.Google	4787
TLS	3445
TLS.Microsoft	488
TLS.YouTube	325
TLS.Amazon	325
TLS.Skype	274
TLS.Cloudflare	182
TLS.Microsoft365	169
TLS.GoogleServices	148
HTTP.WindowsUpdate	66
TLS.Facebook	24
Unknown	20

HTTP.Google	15
TLS.MS_OneDrive	14
HTTP.Microsoft	14
HTTP.Amazon	9
TLS.Twitter	8
HTTP.MS_OneDrive	4
QUIC.Google	4
HTTP.GoogleServices	3
Google	3
QUIC.GoogleServices	3
CiscoVPN.Google	3
TLS.Yahoo	3
TLS.Spotify	2

Πίνακας 2.6 Πίνακας με την συχνότητα εμφάνισης των πρωτόκολλων layer 7 στην δικτυακή κίνηση

Η συντριπτική πλειοψηφία των flows ήταν HTTP, TLS.Google και Google.

(d) Πλήθος bytes των flows που είχαν διεύθυνση προελεύσεως την συγκεκριμένη IP

Τα στατιστικά που προέκυψαν είναι τα παρακάτω:

```
Mean: 1990.89
Max: 6528141.00
Min: 40.00
Standard deviation: 26164.51
Median: 1736.00
```

Παρατηρήθηκε μέσο πλήθος των byte ανα flow 1990.89 bytes, μέγιστο πλήθος bytes σε flow 6.528.141 bytes, ελάχιστο πλήθος bytes σε flow 40 bytes, μέση απόκλιση 26164,51 bytes και διάμεσος 1736 bytes

(e) Πλήθος bytes των flows που είχαν διεύθυνση προορισμού την συγκεκριμένη IP

Τα στατιστικά που προέκυψαν είναι τα παρακάτω:

```
Mean: 27082.14
Max: 13433760.00
Min: 0.00
Standard deviation: 334871.84
Median: 566.00
```

Παρατηρήθηκε μέσο πλήθος των byte ανα flow 27082.14 bytes, μέγιστο πλήθος bytes σε flow 13433760 bytes, ελάχιστο πλήθος bytes σε flow 0 bytes, μέση απόκλιση 334871.84 bytes και διάμεσος 566 bytes

(f) Πλήθος των bytes των flows που είτε παρήχθησαν είτε είχαν ως προορισμό την συγκεκριμένη IP

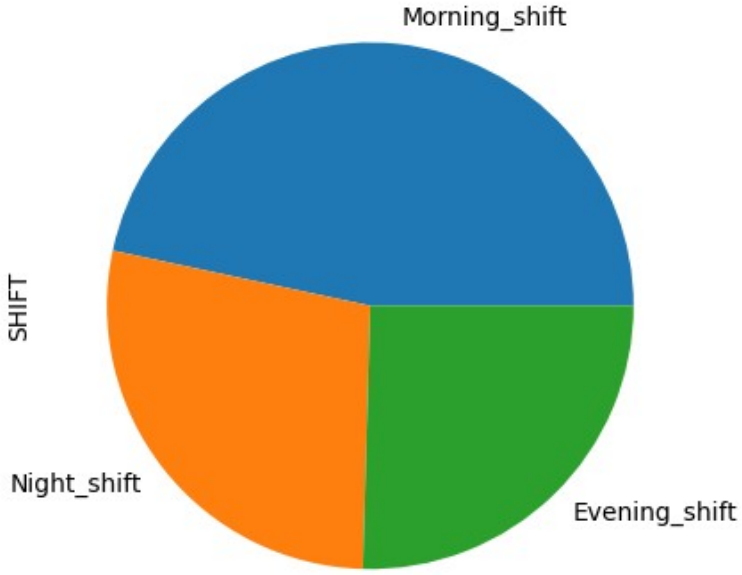
Τα στατιστικά που προέκυψαν είναι τα παρακάτω:

```
Mean: 29073.02
Max: 13713456.00
Min: 40.00
Standard deviation: 341230.36
Median: 2287.00
```

Παρατηρήθηκε μέσο πλήθος των byte ανα flow 29073.02 bytes, μέγιστο πλήθος bytes σε flow 13713456 bytes, ελάχιστο πλήθος bytes σε flow 40 bytes, μέση απόκλιση 341230.36 bytes και διάμεσος 2287 bytes

(g) Βάρδια στην οποία καταγράφηκε το κάθε flow

Σαν σύμβαση για τις βάρδιες χρησιμοποιήθηκε η εξής: οι ώρες ενός 24ωρου, χωρίστηκαν σε τρεις οκτάωρες βάρδιες 00:00-08:00 η πρώτη βάρδια, 08:01-16:00 η δεύτερη και 16:01-23:59 η τρίτη βάρδια.



Εικόνα 2.14 Κυκλικό διάγραμμα των βαρδιών που παρατηρήθηκαν

Τα στατιστικά που προέκυψαν είναι τα παρακάτω:

```
Morning_shift    31392
Night_shift      18793
Evening_shift    17127
```

Παρατηρήθηκαν 31392 flows στην πρώτη βάρδια, 17127 στην δεύτερη βάρδια και 18793 στην τρίτη.

Κεφάλαιο 4

Υλοποίηση Συστήματος

4.1 Εργαλεία

4.1.1 Python

Για την ανάπτυξη του λογισμικού χρησιμοποιήθηκε η γλώσσα προγραμματισμού python. Η python είναι υψηλού επιπέδου γλώσσα και υποστηρίζει τόσο διαδικαστικό (procedural) όσο και αντικειμενοστραφές (object-oriented) προγραμματισμό. Δημιουργήθηκε το 1989 από τον Ολλανδό Guido van Rossum στο ερευνητικό κέντρο Centrum Wiskunde & Informatica και κυκλοφόρησε το 1991. Βασικός στόχος της είναι η ευκολία χρήσης της, το απλό συντακτικό και η αναγνωσιμότητα του κώδικα [45] [46]. Παρέχει πλειάδα βιβλιοθηκών (packages) που την καθιστούν κατάλληλη για μεγάλη γκάμα προγραμματιστικών εφαρμογών. Για την ανάπτυξη του Behaviour Simulator μεταξύ άλλων χρησιμοποιήθηκαν τα παρακάτω packages: connexion, flask, multiprocessing, threading, selenium, json, numpy κ.α.

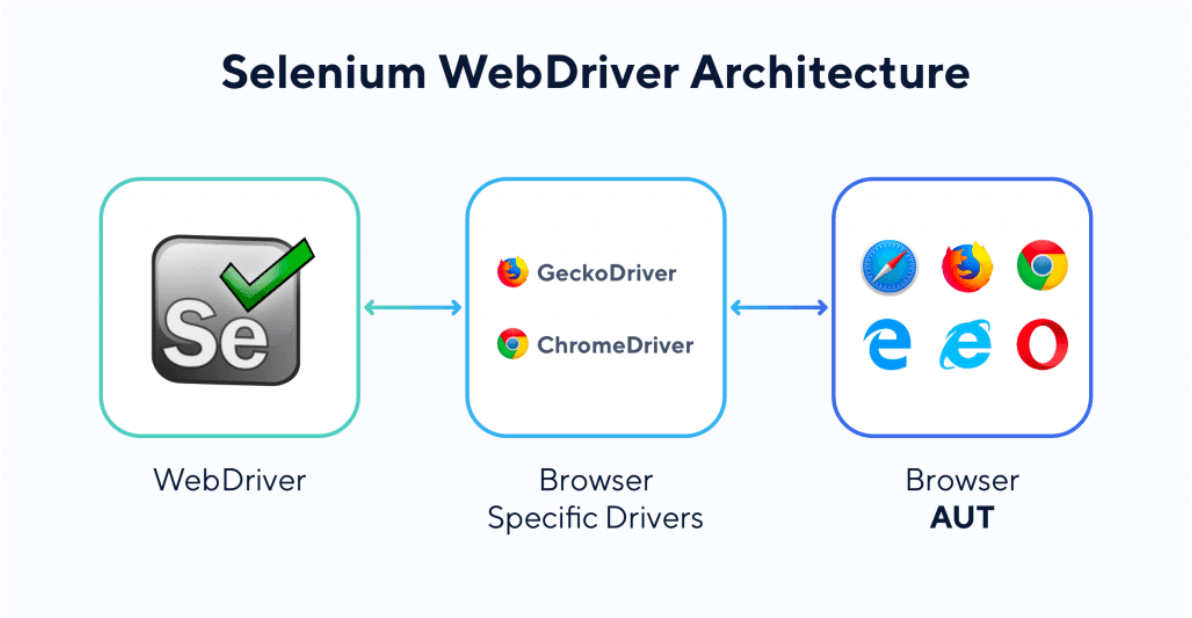
Η πρώτη δημοφιλής ιστορικά έκδοση της, η Python 2.0 κυκλοφόρησε το 2000, ενώ το 2008 κυκλοφόρησε η Python 3.0. Η έκδοση που χρησιμοποιήθηκε στην παρούσα εργασία είναι η 3.6.9.

4.1.2 Τεχνολογίες Αυτοματισμού

Παρακάτω παρατίθενται συνοπτικά οι τεχνολογίες που μπορούν να αξιοποιηθούν στην ανάπτυξη λογισμικού με σκοπό την προσομοίωση της ανθρώπινης δικτυακής συμπεριφοράς.

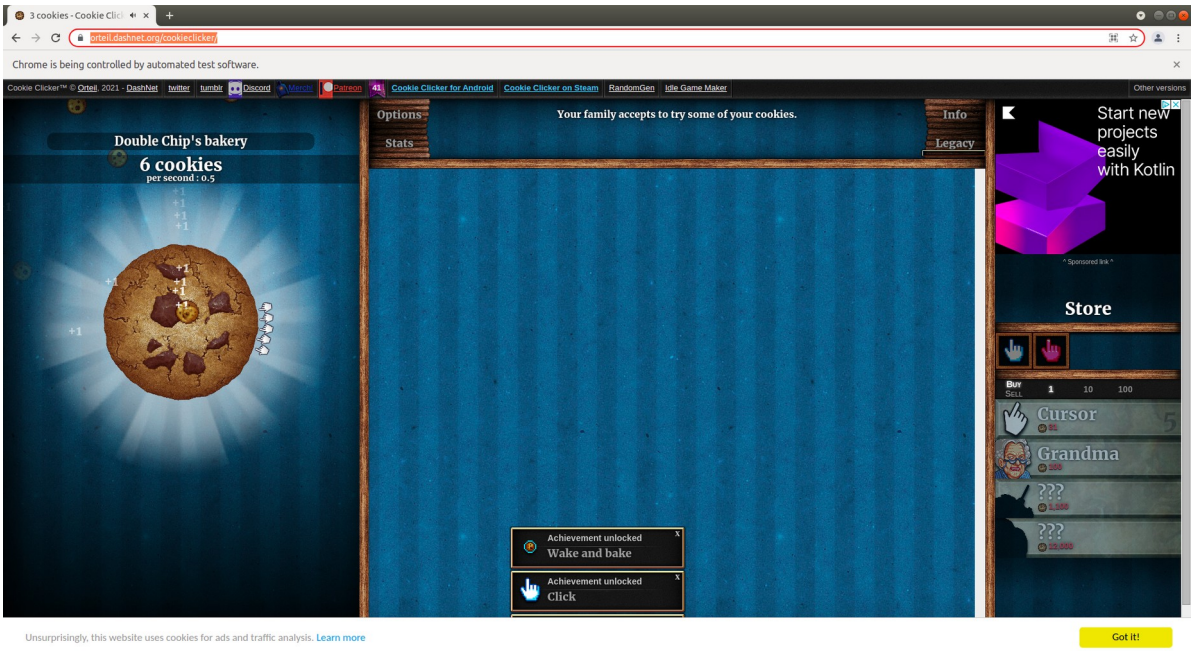
4.1.2.1 Selenium WebDriver – Αυτοματισμός Browser

Το Selenium είναι ένα open-source λογισμικό που χρησιμοποιείται για δοκιμές (tests) σε διαδικτυακές εφαρμογές που ανοίγουν σε browser και για αυτοματοποίηση εργασιών επίσης σε browser. Δημιουργήθηκε από τον Jason Huggins το 2004. Αποτελείται από ένα σύνολο διαφορετικών εργαλείων λογισμικού, το καθένα με διαφορετική προσέγγιση για την αυτοματοποίηση των δοκιμών. Το πιο δημοφιλές (και αυτό που χρησιμοποιήθηκε στην παρούσα εργασία) είναι το Selenium WebDriver. Μέσω κατάλληλα διαμορφωμένων APIs ο webdriver δέχεται εντολές τις οποίες επικοινωνεί απευθείας στον εκάστοτε browser (υποστηρίζεται από Mozilla Firefox, Google Chrome, Opera, Microsoft Edge, Internet Explorer κ.α.). Επιτρέπει τον εντοπισμό στοιχείων στο UI των ιστοσελίδων/δικτυακών εφαρμογών και άρα την σύγκριση των αναμενόμενων αποτελεσμάτων των δοκιμών με την πραγματική συμπεριφορά της εφαρμογής. [47] [48]



Εικόνα 4.1 Η Αρχιτεκτονική του Selenium WebDriver[25]

Στο Παράρτημα Κώδικα Κ.1 παρατίθεται ενδεικτικός κώδικας με τον οποίο αυτοματοποιείται το online browser παιχνίδι cookie clicker[26].



Εικόνα 4.2 Το online browser παιχνίδι cookie clicker ενώ ελέγχεται αυτοματοποιημένα μέσω python script

25 [Πηγή: https://www.katalon.com/resources-center/blog/selenium-alternative-solution/]
26 [https://orteil.dashnet.org/cookieclicker/]

4.1.2.2 Αυτοματισμός Skype

Η βιβλιοθήκη της python skpy επιτρέπει την αλληλεπίδραση με το Skype HTTP API. Υποστηρίζει μεταξύ άλλων την διαχείριση της λίστας επαφών ενο χρήστη, την αποστολή και λήψη μηνυμάτων, εικόνων, αρχείων γενικότερα, την δημιουργία ομαδικών συζητήσεων και την διαχείριση τους, αλλαγή των ρυθμίσεων κ.α. [49]

Στο παράρτημα Κώδικα Κ.2 παρατίθεται ενδεικτικός κώδικας όπου αρχικά επιχειρείται σύνδεση στο skype, έπειτα ανάγνωση παλαιών συνομιλίων και τέλος αποστολή μηνύματος σε τυχαία επαφή.

4.1.2.3 Αυτοματισμός SSH

Η βιβλιοθήκη της python paramiko επιτρέπει την αυτοματοποίηση της επικοινωνίας και της μεταφοράς αρχείων μέσω του πρωτόκολλου SSH. Σύμφωνα με το paramiko.org η βιβλιοθήκη επιτρέπει την προσομοίωσή της λειτουργικότητας τόσο από την πλευρά του server όσο και από την πλευρά του client. Σαν client επιχειρείται authentication μέσω κωδικού ή κλειδιού, το «τρέξιμο» εντολών, η λήψη και η αποστολή αρχείων [50] [51].

Στο παράρτημα Κώδικα Κ.3 παραρατίθεται ενδεικτικός κώδικας όπου μετά από εγκατάσταση σύνδεσης με απομακρυσμένο υπολογιστή, «τρέχουν» ορισμένες bash εντολές.

4.1.2.4 Αυτοματισμός FTP

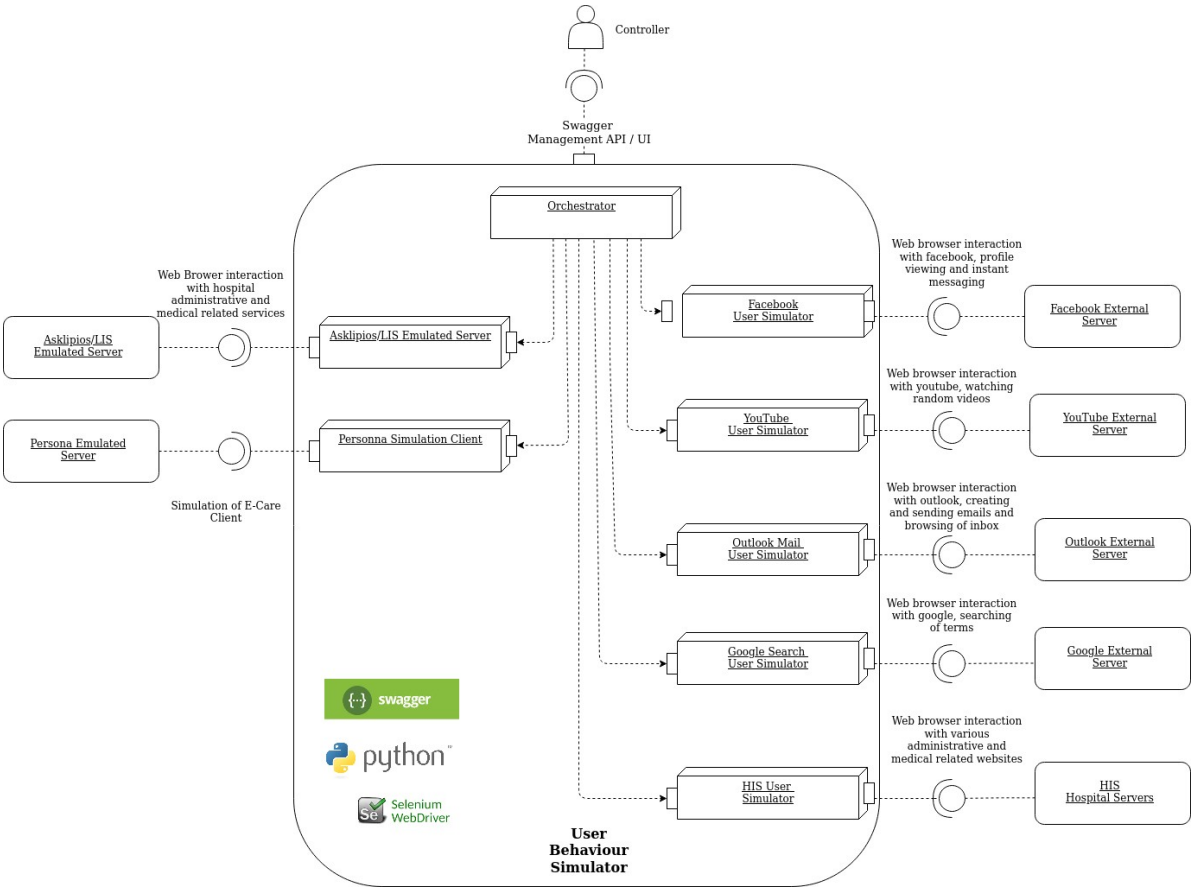
Η βιβλιοθήκη της python ftplib επιτρέπει την αυτοματοποίηση της μεταφοράς αρχείων μέσω του πρωτόκολλου FTP. [52]

Στο παράρτημα Κώδικα Κ.4 παρατίθεται ενδεικτικός κώδικας όπου αρχικά επιχειρείται σύνδεση στην διεύθυνση ftp1.at.proftpd.org και στη συνέχεια λαμβάνεται τοπικά ένα αρχείο.

4.2 Περιγραφή του Behaviour Simulator

4.2.1 Σκοπός του Behaviour Simulator

Σκοπός του Behaviour Simulator που θα παρουσιαστεί παρακάτω αναλυτικά, είναι η προσομοίωση της ψηφιακής συμπεριφοράς, του υγειονομικού και μη, προσωπικού ενός νοσοκομείου με τελικό στόχο την δημιουργία σετ δεδομένων (dataset) με καλόβουλη δικτυακή κίνηση νοσοκομείου. Το tool επιτρέπει την δημιουργία “προφίλ” παραμετροποιήσιμων ως προς την διαδικτυακή συμπεριφορά τους, ώστε να μπορούν να προσομοιωθούν ικανοποιητικά ρόλοι όπως του γιατρού, του διοικητικού προσωπικού κλπ.



Εικόνα 4.3 Διάγραμμα αρχιτεκτονικής του Behaviour Simulator

4.2.2 Παρουσίαση των προφίλ

Στην εφαρμογή έχουν ενσωματωθεί δύο προφίλ, το Custom Profile και το Quick Profile.
Και τα δύο προφίλ ουσιαστικά παρέχουν την ίδια λειτουργικότητα και μπορούν να προσομοίωσουν τις ίδιες συμπεριφορές με μικρές διαφοροποιήσεις που επεξηγούνται σε παρακάτω παραγράφους.

Οι κοινές λειτουργίες που παρέχουν τα δύο προφίλ είναι:

i. Εκκίνηση μιας σειράς από «εφαρμογές», στις οποίες γίνεται προσομοίωση της επικοινωνίας του προφίλ με τον client (front end) του εκάστοτε service. Χαρακτηριστικά παραδείγματα είναι δημοφιλείς ιστοσελίδες όπως το Facebook, το Youtube, το Outlook αλλά και ιστοσελίδες τις οποίες επισκέπτεται ένας γιατρός κατά την ώρα της εργασίας του, όπως το διαδικτυακό φαρμακείο Γαληνός (https://www.galinos.gr/), η ιστοσελίδα της ηλεκτρονικής συνταγογράφησης (https://www.e-prescription.gr) κ.α ή ιστοσελίδες που αντίστοιχα επίσκεπτεται ένας διοικητικός υπάλληλος κατά την ώρα της εργασίας του όπως η Γενική Γραμματεία Πληροφοριακών Συστημάτων Δημόσιας Διοίκησης (https://www.gsis.gr), η ιστοσελίδα του ΕΟΠΥΥ (https://www.eopyy.gov.gr) κ.α. Η επιλογή των εφαρμογών και των ιστοσελίδων που είναι διαθέσιμες δεν έγινε τυχαία αλλά είναι αποτέλεσμα στατιστικής ανάλυσης αληθινής κίνησης νοσοκομείου. Οι παραπάνω εφαρμογές παρατίθενται αναλυτικά στην παράγραφο 4.2.3.1

ii. Εκκίνηση του ενδονοσοκομειακού συστήματος Ασκληπιός – LIS. Το εν λόγω σύστημα είναι πιστό αντίγραφο του αντίστοιχου πραγματικού, και ουσιαστικά «καταγράφει» την συνολική πορεία του ασθενή, από την άφιξη του στο νοσοκομείο και την εισαγωγή των στοιχείων του στην αντίστοιχη καρτέλα ασθενούς, μέχρι και ενδιάμεσες διαδικασίες όπως καταγραφή εξετάσεων που του έγιναν και των αποτελεσμάτων τους, την παρακολούθηση του ιστορικού του, την εισαγωγή του σε κλίνη κλπ, έως και τελικά την εξαγωγή του. Αποτελείται από τρία υποσυστήματα, εκ των οποίων τα δύο τελευταία έχουν αυτοματοποιηθεί:
   a) Διοικητικό: Διαχειρίζεται την εισαγωγή ενός ασθενούς
   b) Ασκληπιός: Αποτελεί το κεντρικό νοσοκομειακό σύστημα και διαζειρίζεται τις διάφορες διαδικασίες της νοσηλείας ενός ασθενούς.
   c) LIS: Διαχειρίζεται τις εξετάσεις ενός ασθενούς, από την πλευρά του εργαστηρίου.

   Αναλυτική περιγραφή του συστήματος και των σεναρίων του παρατίθενται στην παράγραφο 4.2.3.2

iii. Εκκίνηση του συστήματος της persona. Το εν λόγω σύστημα δημιουργεί χρήστες (users) για τους οποίους αποστέλλει πληροφορίες σε έναν server ανα τακτά χρονικά διαστήματα. Οι πληροφορίες που αποστέλλονται είναι ιατρικής φύσεως, και αποτελούν χαρακτηριστικούς δείκτες της υγείας ενός ασθενούς όπως ο καρδιακός ρυθμός, η διαστολική και συστολική πίεση του αίματος κ.α. Μια περιγραφή του συστήματος και των σεναρίων του παρατίθενται αναλυτικά στην παράγραφο 4.2.3.3

4.2.2.1 Quick Profile

Σκοπός του Quick Profile είναι η «γρήγορη» εκκίνηση ενός προφίλ, καθώς εκκινείται στο UI του swagger API χωρίς να απαιτείται να αποσταλλεί περαιτέρω πληροφορία. Η τελευταία είναι αποθηκευμένη στο αρχείο quick_profile_config.json στην πλευρά του server. Το περιεχόμενο του εν λόγω αρχείου όπως και οδηγίες για την παραμετροποίηση συνολικά του Quick Profile υπάρχουν στην παράγραφο 4.5.4.2

4.2.2.2 Custom Profile

Το Custom Profile παρέχει την δυνατότητα καθορισμού της αναλύτικης περιγραφής της συμπεριφοράς που θα ακολουθηθεί από το προφίλ, μέσω του request body του αιτήματος, που θα αποσταλλεί από τον χρήστη. Στον αντίποδα της «γρήγορης» εκκίνησης ενός προφίλ, που παρέχει το Quick profile, ο χρήστης μπορεί απομακρυσμένα να καθορίσει την συνολική διάρκεια της προσομοίωσης, τις εφαρμογές που θα «τρέξουν», πόσες φορές θα «τρέξουν»και τις αντίστοιχες διάρκειες κάθε εφαρμογής, αλλά και τυχών ενδιάμεσα διαστήματα αδράνειας ανάμεσα στην επανεκκίνηση τους. Αναλυτικές οδηγίες για την παραπάνω παραμετροποίηση δίνονται στην παράγραφο 4.5.4.1

4.2.3 Λειτουργίες

4.2.3.1 Εφαρμογές που ανοίγουν μέσω Browser

Όνομα Εφαρμογής	**URL**	**Περιγραφή Σεναρίου**
Google	https://www.google.com/	ΒΗΜΑΤΑ
		1.Καθορισμός μέσω του αρχείου google_search_queries.txt των όρων που θα αναζητηθούν στην μηχανή αναζήτησης. Σε κάθε αναζήτηση ο όρος που θα αναζητηθεί επιλέγεται τυχαία. Σχόλιο: το αρχείο google_search_queries.txt βρίσκεται υπό το σχετικό directory *API*/profile_functions/google/ Οι όροι προς αναζήτηση δεν χρειάζεται να είναι σε κάποια συγκεκριμένη σειρά, πέρα από το ότι, κάθε ένας βρίσκεται σε νέα γραμμή εντός του αρχείου. **Επαφίεται στον χρήστη να εισάγει όρους προς αναζήτηση καθώς υπάρχουν μόνο τρεις όροι, εν είδει δείγματος.**
		2.Μετάβαση στην ιστοσελίδα: https://www.google.com/
		3.Επιλογή του κουμπιού “I Agree” στο pop-up της google για τα cookies.
		4.Τυχαία επιλογή όρου αναζήτησης. Εισαγωγή του όρου

		στην φόρμα αναζήτησης και αποστολή του πλήκτρου “Enter”. 5.Επανάληψη του βήματος 4 μέχρι το τέλος της χρονικής διάρκειας του σεναρίου.
Youtube	https://www.youtube.com/	ΒΗΜΑΤΑ 1.Μετάβαση στην ιστοσελίδα: https://www.youtube.com/ 2.Επιλογή του κουμπιού “I Agree” στο pop-up της google για τα cookies. 3.Σύνδεση στο youtube με εισαγωγή ονόματος χρήστη και αντίστοιχου κωδικού. Κατόπιν αποστολή του πλήκτρου “Enter”. Σχόλιο: υπό το directory API/profile_functions/youtube/ πρέπει να βρίσκεται ένα αρχείο .env, που θα περιλαμβάνει το username του χρήστη και τον κωδικό του χρήστη. Τα στοιχεία δεν χρειάζεται να είναι σε κάποια συγκεκριμένη σειρά, πέρα από το ότι κάθε ένα βρίσκεται σε νέα γραμμή εντός του αρχείου. **Επαφίεται στον χρήστη να εισάγει τα παραπάνω στοιχεία.** Παράδειγμα περιεχομένου του .env: EMAIL="xxxxx@xxxx.xxx" PASSWORD="XXXXXXXXXX" 4.Μετάβαση στην σελίδα με τα trending videos (https://www.youtube.com/feed/trending), δημιουργία λίστας με τα trending videos. Επιλογή ενός τυχαίου βίντεο. Κλείσιμο διαφημίσεων αν εμφανιστούν. Αναπαραγωγή του βίντεο. 5.Επανάληψη του βήματος 4 μέχρι το τέλος της χρονικής διάρκειας του σεναρίου.
Outlook	https://outlook.live.com/owa/	ΒΗΜΑΤΑ 1.Μετάβαση στην ιστοσελίδα: https://outlook.live.com/owa/ 2.Σύνδεση στο outlook με εισαγωγή ονόματος χρήστη και αντίστοιχου κωδικού. Κατόπιν αποστολή του πλήκτρου “Enter”. Σχόλιο: υπό το directory API/profile_functions/outlook/ πρέπει να βρίσκεται ένα αρχείο .env, που θα περιλαμβάνει το email του χρήστη, τον κωδικό του χρήστη όπως και το

		email του παραλήπτη. Τα στοιχεία δεν χρειάζεται να είναι σε κάποια συγκεκριμένη σειρά, πέρα από το ότι κάθε ένα να βρίσκεται σε νέα γραμμή εντός του αρχείου. **Επαφίεται στον χρήστη να εισάγει τα παραπάνω στοιχεία.** Παράδειγμα περιεχομένου του .env: OUTLOOK_EMAIL="xxxxx@xxxx.xxx" OUTLOOK_PASSWORD="xxxxxxxxxxxxxxx" RECEIPTER_EMAIL="xxxxx@xxxx.xxx"
		3.Επιλογή “No”/”Όχι” στο pop up “Stay signed in?”/”Θέλετε να παραμείνετε συνδεδεμένος;”
		4.Πιθανότητα για περιήγηση στα ληφθέντα email στον φάκελο Inbox. Δημιουργία λίστας με όλα τα ληφθέντα email. Επιλογή τυχαίου email από την λίστα και άνοιγμα του.
		5.Πιθανότητα για δημιουργία νέου email και αποστολή του. Επιλογή “New Message”. Συμπλήρωση του email του παραλήπτη στην αντίστοιχη φόρμα. Συμπλήρωση του πεδίου “Subject” με τυχαίους χαρακτήρες. Συμπλήρωση του κυρίως σώματος του email με τυχαίους χαρακτήρες. Επιλογή “Send”.
		6.Επανάληψη των βημάτων 4 και 5 μέχρι το τέλος της χρονικής διάρκειας του σεναρίου.
Facebook	https://el-gr.facebook.com/	ΒΗΜΑΤΑ 1.Μετάβαση στην ιστοσελίδα: https://el-gr.facebook.com/
		2.Επιλογή του κουμπιού “I Agree” στο pop-up για τα cookies (αν εντοπιστεί).
		3.Σύνδεση στο facebook με εισαγωγή ονόματος χρήστη και αντίστοιχου κωδικού. Κατόπιν αποστολή του πλήκτρου “Enter”. Σχόλιο: υπό το directory API/profile_functions/facebook/ πρέπει να βρίσκεται ένα αρχείο .env, που θα περιλαμβάνει το username του χρήστη και τον κωδικό του χρήστη. Τα στοιχεία δεν χρειάζεται να είναι σε κάποια συγκεκριμένη σειρά, πέρα από το ότι κάθε ένα βρίσκεται σε νέα γραμμή εντός του αρχείου. **Επαφίεται στον χρήστη να εισάγει τα παραπάνω στοιχεία.**

		Παράδειγμα περιεχομένου του .env: FACEBOOK_EMAIL1="xxxxx@xxxx.xxx" FACEBOOK_PASSWORD1="XXXXXXXXXX" 4.Επιλογή του κουμπιού “I Agree” στο pop-up για τα cookies (αν εντοπιστεί). 5.Πιθανότητα για περιήγηση στην ιστοσελίδα. Αρχικά καθορισμός μέσω του αρχείου facebook_search_queries.txt των όρων που θα αναζητηθούν στην μηχανή αναζήτησης του facebook. Σε κάθε αναζήτηση ο όρος που θα αναζητηθεί επιλέγεται τυχαία. Αναζήτηση του επιλεχθέντα όρου και επιλογή τυχαίου συνδέσμου (link) από τα αποτελέσματα της αναζήτησης. Σχόλιο: το αρχείο facebook_search_queries.txt βρίσκεται υπό το σχετικό directory *API*/profile_functions/facebook/ Οι όροι προς αναζήτηση δεν χρειάζεται να είναι σε κάποια συγκεκριμένη σειρά, πέρα από το ότι, κάθε ένας βρίσκεται σε νέα γραμμή εντός του αρχείου. **Επαφίεται στον χρήστη να εισάγει όρους προς αναζήτηση καθώς υπάρχουν μόνο κάποιοι λίγοι όροι εν είδει δείγματος.** 6.Πιθανότητα για άνοιγμα της εφαρμογής του messenger και αποστολή μηνύματος σε κάποια έπαφη. Επιλογή του εικονιδίου του messenger και κατόπιν επιλογή από το εμφανιζόμενο υπομενού “Εμφάνιση όλων στο Messenger”. Δημιουργία λίστας με όλες τις επαφές. Επιλογή τυχαία επαφής. Συμπλήρωση της φόρμας μηνύματος με τυχαίους χαρακτήρες. Αποστολή του πλήκτρου “Enter”. Σχόλιο: **Επαφίεται στον χρήστη να έχει κάνει “add” επαφών/φίλων χειρονακτικά στο κάθε facebook προφίλ που θα δημιουργήσει ή να δώσει στοιχεία χρήστη facebook που έχει ήδη επαφές.** 7.Επανάληψη των βημάτων 6 και 7 μέχρι το τέλος της χρονικής διάρκειας του σεναρίου.
Galinos	https://www.galinos.gr/	ΒΗΜΑΤΑ 1.Μετάβαση στην ιστοσελίδα: https://www.galinos.gr/ 2.Επιλογή τυχαίου συνδέσμου (link).

		3.Επανάληψη του βήματος 2 μέχρι το τέλος της χρονικής διάρκειας του σεναρίου.
Promitheus	http:// www.eprocurement.gov.gr	ΒΗΜΑΤΑ 1.Μετάβαση στην ιστοσελίδα: http://www.eprocurement.gov.gr 2.Επιλογή τυχαίου συνδέσμου (link). 3.Επανάληψη του βήματος 2 μέχρι το τέλος της χρονικής διάρκειας του σεναρίου.
Apografi_http	http://apografi.gov.gr/	ΒΗΜΑΤΑ 1.Μετάβαση στην ιστοσελίδα: http://apografi.gov.gr/ 2.Επιλογή τυχαίου συνδέσμου (link). 3.Επανάληψη του βήματος 2 μέχρι το τέλος της χρονικής διάρκειας του σεναρίου.
Gsis	https://www.gsis.gr	ΒΗΜΑΤΑ 1.Μετάβαση στην ιστοσελίδα: https://www.gsis.gr 2.Επιλογή τυχαίου συνδέσμου (link). 3.Επανάληψη του βήματος 2 μέχρι το τέλος της χρονικής διάρκειας του σεναρίου.
Idika	https://www.idika.gr	ΒΗΜΑΤΑ 1.Μετάβαση στην ιστοσελίδα: https://www.idika.gr 2.Επιλογή τυχαίου συνδέσμου (link). 3.Επανάληψη του βήματος 2 μέχρι το τέλος της χρονικής διάρκειας του σεναρίου.
Ebaby	https://ebaby.ypes.gr/	ΒΗΜΑΤΑ 1.Μετάβαση στην ιστοσελίδα: https://ebaby.ypes.gr/ 2.Επιλογή τυχαίου συνδέσμου (link). 3.Επανάληψη του βήματος 2 μέχρι το τέλος της χρονικής διάρκειας του σεναρίου.

Eopyy	https://www.eopyy.gov.gr	ΒΗΜΑΤΑ 1.Μετάβαση στην ιστοσελίδα: https://www.eopyy.gov.gr 2.Επιλογή τυχαίου συνδέσμου (link). 3.Επανάληψη του βήματος 2 μέχρι το τέλος της χρονικής διάρκειας του σεναρίου.
E_prescription	https://www.e-prescription.gr	ΒΗΜΑΤΑ 1.Μετάβαση στην ιστοσελίδα: https://www.e-prescription.gr 2.Επιλογή τυχαίου συνδέσμου (link). 3.Επανάληψη του βήματος 2 μέχρι το τέλος της χρονικής διάρκειας του σεναρίου.
Diavgeia	https://www.diavgeia.gov.gr	ΒΗΜΑΤΑ 1.Μετάβαση στην ιστοσελίδα: https://www.diavgeia.gov.gr 2.Επιλογή τυχαίου συνδέσμου (link). 3.Επανάληψη του βήματος 2 μέχρι το τέλος της χρονικής διάρκειας του σεναρίου.
E_Services	https://eservices.yeka.gr/	ΒΗΜΑΤΑ 1.Μετάβαση στην ιστοσελίδα: https://eservices.yeka.gr/ 2.Επιλογή τυχαίου συνδέσμου (link). 3.Επανάληψη του βήματος 2 μέχρι το τέλος της χρονικής διάρκειας του σεναρίου.
Dypethessaly	https://www.dypethessaly.gr	ΒΗΜΑΤΑ 1.Μετάβαση στην ιστοσελίδα: https://www.dypethessaly.gr 2.Επιλογή τυχαίου συνδέσμου (link). 3.Επανάληψη του βήματος 2 μέχρι το τέλος της χρονικής διάρκειας του σεναρίου.

Πίνακας 4.1 Εφαρμογές που ανοίγουν μέσω browser

4.2.3.2 Ασκληπιός - LIS

Το εν λόγω σύστημα είναι πιστό αντίγραφο αντίστοιχου πραγματικού ενδονοσοκομειακού, και επιτελεί λειτουργίες όπως η καταγραφή της συνολικής πορείας ενός ασθενή, την εισαγωγή των στοιχείων του στην αντίστοιχη καρτέλα ασθενούς, μέχρι και ενδιάμεσες διαδικασίες όπως καταγραφή εξετάσεων που του έγιναν και των αποτελεσμάτων τους, την παρακολούθηση του ιστορικού του, την εισαγωγή του σε κλίνη κλπ, έως και τελικά την εξαγωγή του.

Αποτελείται από τρια υποσυστήματα:

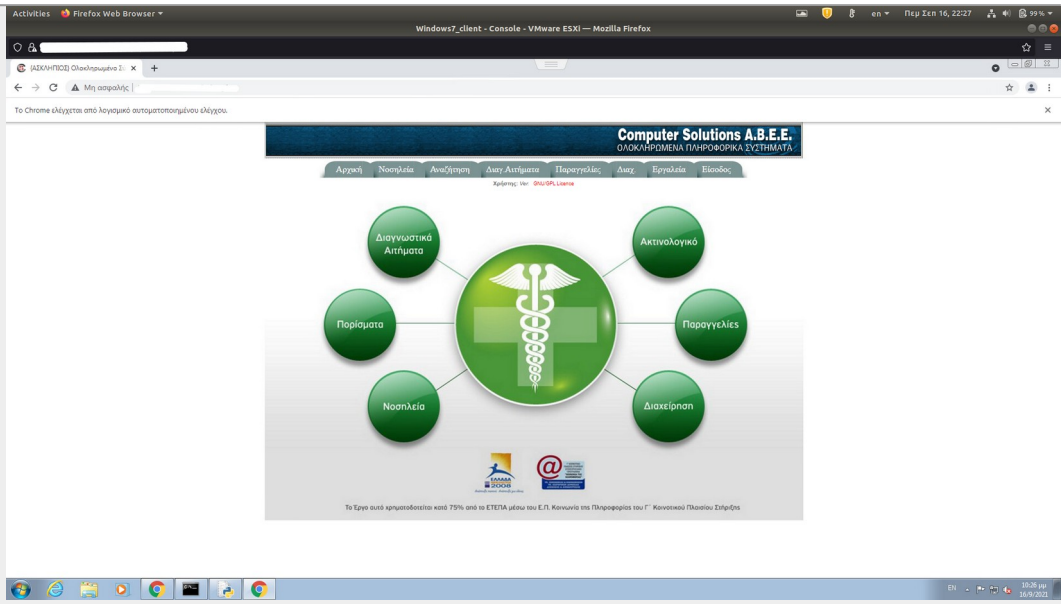
i. Διοικητικό: Διεκπεραιώνει την εισαγωγή ενός ασθενούς στο νοσοκομείο. Αρχικά, καταχωρεί τα στοιχεία του στο μητρώο αν δεν έχει ξανανοσηλευτεί ή εναλλακτικά τον εντοπίζει στο μητρώο. Κατόπιν τον εισάγει στο νοσοκομείο. Το υποσύστημα αυτό δεν έχει αυτοματοποιηθεί στα πλαίσια αυτής της εργασίας, καθώς δεν «τρέχει» σε browser. Η διαδικασία της εισαγωγής γίνεται από τον χρήστη χειροκίνητα εξαρχής ώστε να μπορούν να εκκινηθούν και τα δύο παρακάτω υποσυστήματα.
ii. Ασκληπιός: Πρόκειται για το κεντρικό σύστημα του νοσοκομείου. Συμπεριλαμβάνει μεταξύ άλλων λειτουργίες όπως η διαχείριση της νοσηλείας των ασθενών (πχ εισαγωγή σε κλίνη, εξαγωγή κ.α.), παραγγελίες φαρμάκων και νοσοκομειακών υλικών, ευρετήριο διαγνωστικών αιτημάτων, αναζήτηση ασθενούς κ.α. Το σύστημα που χρησιμοποιείται στα πλαίσια της προσομοίωσης έχει μειωμένες δυνατότητες σε σχέση με το πραγματικό, όπως για παράδειγμα, μία μόνο διαθέσιμη κλινική την παθολογική κλπ.

iii. LIS: Διεκπαιρεώνει τις λειτουργίες των εργαστηρίων ενός νοσοκομείου. Δίνει την δυνατότητα αναζήτησης παλαιότερων εξετάσεων ενός ασθενή, καταχώρησης των αποτελεσμάτων νέων εξετάσεων κ.α.

Σημειώνεται ότι υπάρχει η δυνατότητα η προσομοίωση να επαναλαμβάνει το σενάριο του υποσυστήματος του Ασκληπιού, χωρίς να προχωράει στο αντίστοιχο υποσύστημα του LIS (βλ. 4.5.4.1.8).

Το σενάριο του Ασκληπιός – LIS συστήματος είναι το παρακάτω:

ΒΗΜΑΤΑ	ΣΕΝΑΡΙΟ
1	Μετάβαση στην σελίδα του Ασκληπιού.



2 Επιλογή του tab “Είσοδος”.

3 Συμπλήρωση των πεδίων “όνομα” και “κωδικός” και κατόπιν επιλογή “Συνέχεια”.

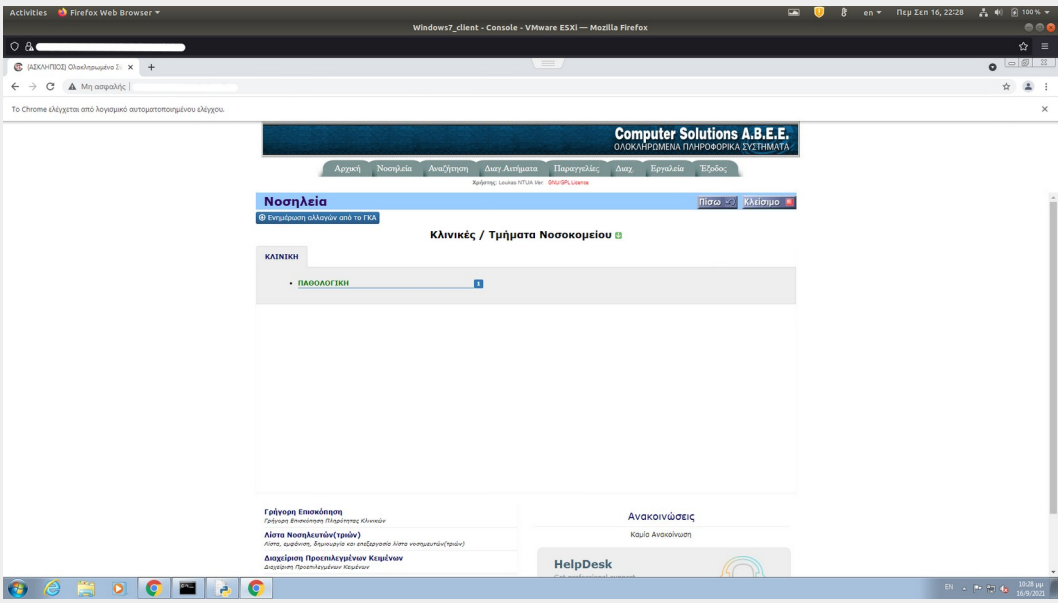
Σχόλιο: υπό το directory API/asklipios_LIS πρέπει να βρίσκεται ένα αρχείο .env, που θα περιλαμβάνει το username του χρήστη και τον κωδικό του χρήστη. Τα στοιχεία δεν χρειάζεται να είναι σε κάποια συγκεκριμένη σειρά, πέρα από το ότι κάθε ένα βρίσκεται σε νέα γραμμή εντός του αρχείου. Τα ίδια στοιχεία χρησιμοποιούνται και για την σύνδεση LIS.

Επαφίεται στον χρήστη να εισάγει τα παραπάνω στοιχεία.

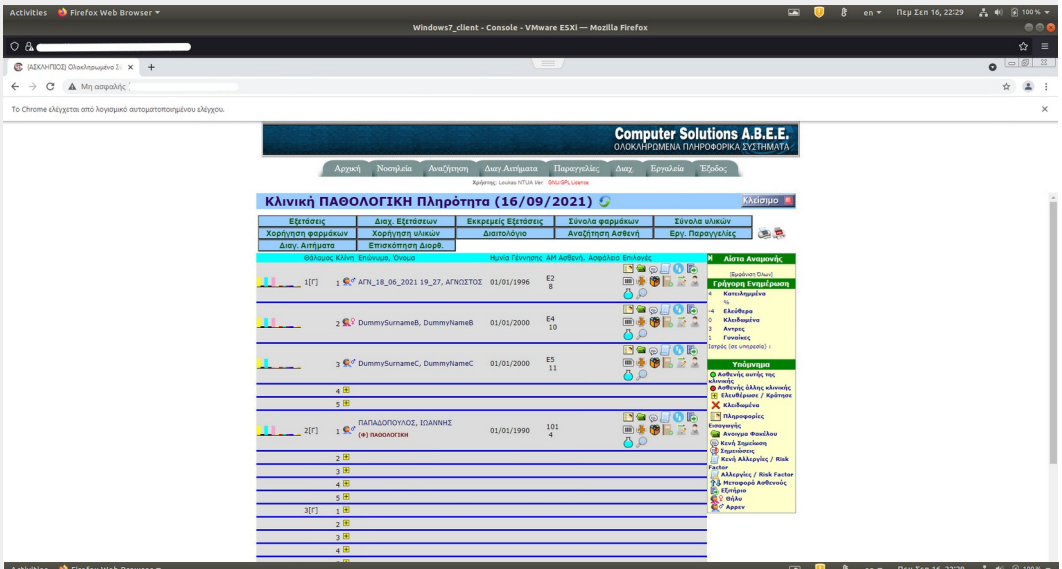
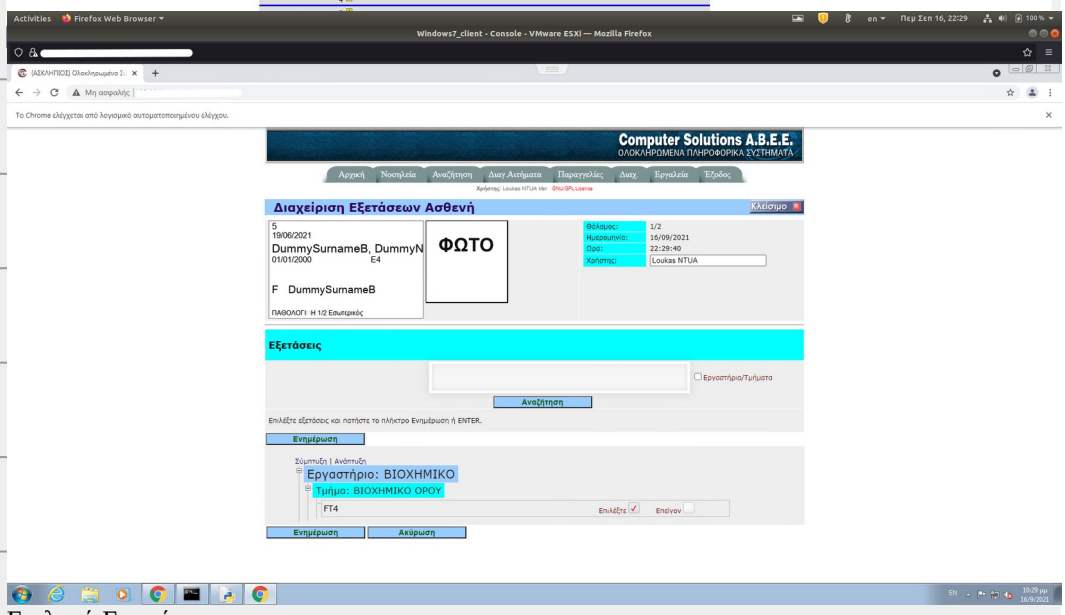
Παράδειγμα περιεχομένου του .env:

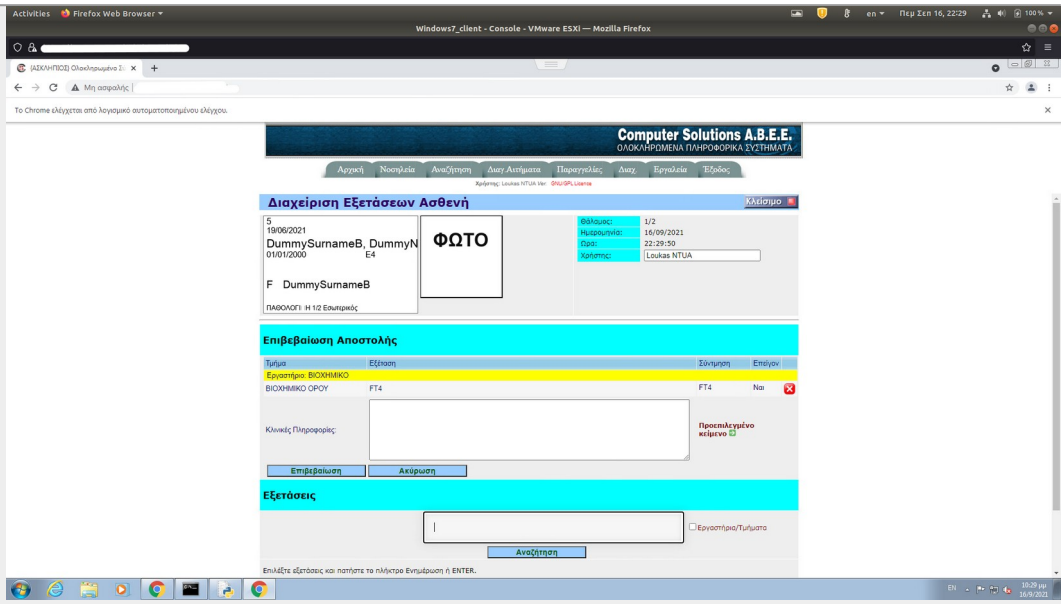
USERNAME_LIS="XXXXXXXXX"
PASSWORD_LIS="XXXXXXXXXX"

4 Επιλογή της κλινικής “ΠΑΘΟΛΟΓΙΚΗ”.

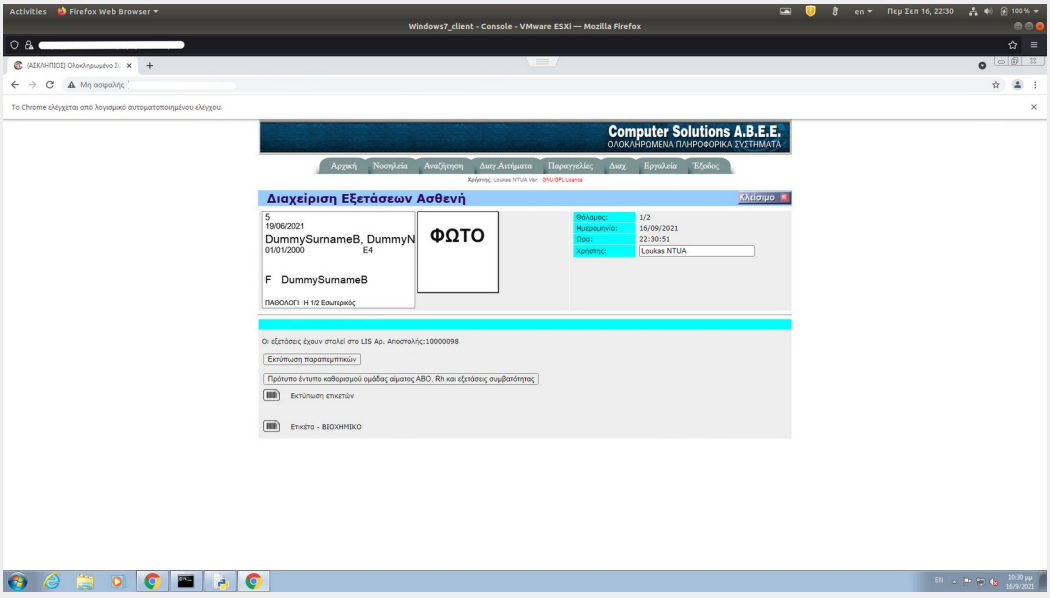


5 Εισαγωγή νέου ασθενούς από την “Λίστα Αναμονής”, αν υπάρχει κάποιος διαθέσιμος αλλιώς μετάβαση στο βήμα 8.

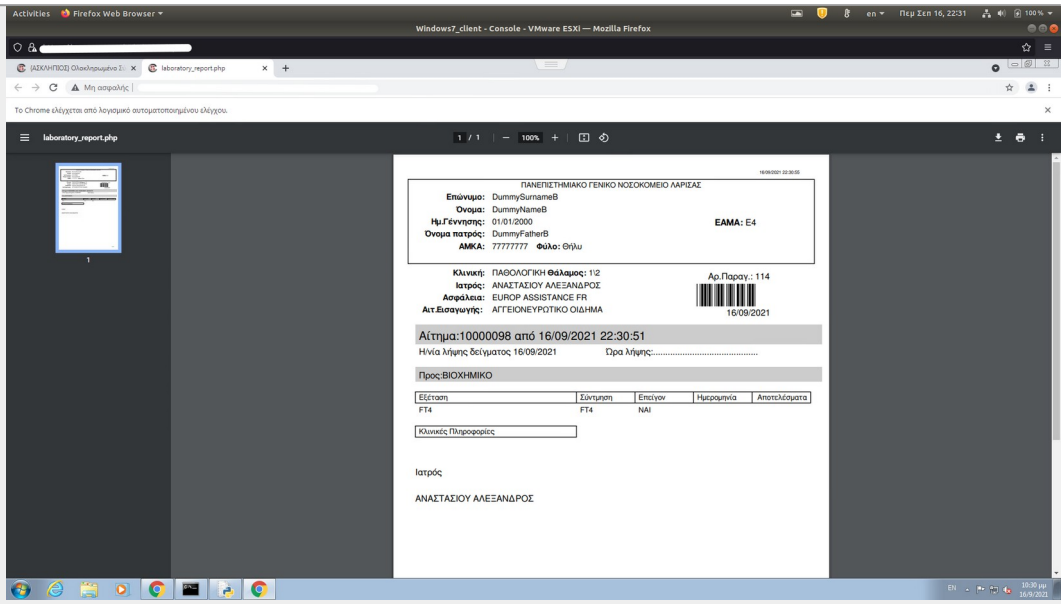
6	Επιλογή διαθέσιμης κλίνης, αν υπάρχει αλλιώς μετάβαση στο βήμα 8.
7	Καταχώρηση του ασθενούς στην κλίνη.
8	Επιλογή του εικονιδίου με το μπλε μπουκάλι για καταχώρηση νέων εξετάσεων, αν υπάρχει κάποιο διαθέσιμο αλλιώς μετάβαση στο βήμα 32.
9	
10	
11	
12	
13	
14	 Επιλογή Ενημέρωση.
15	Επιλογή “Επιβεβαίωση”.

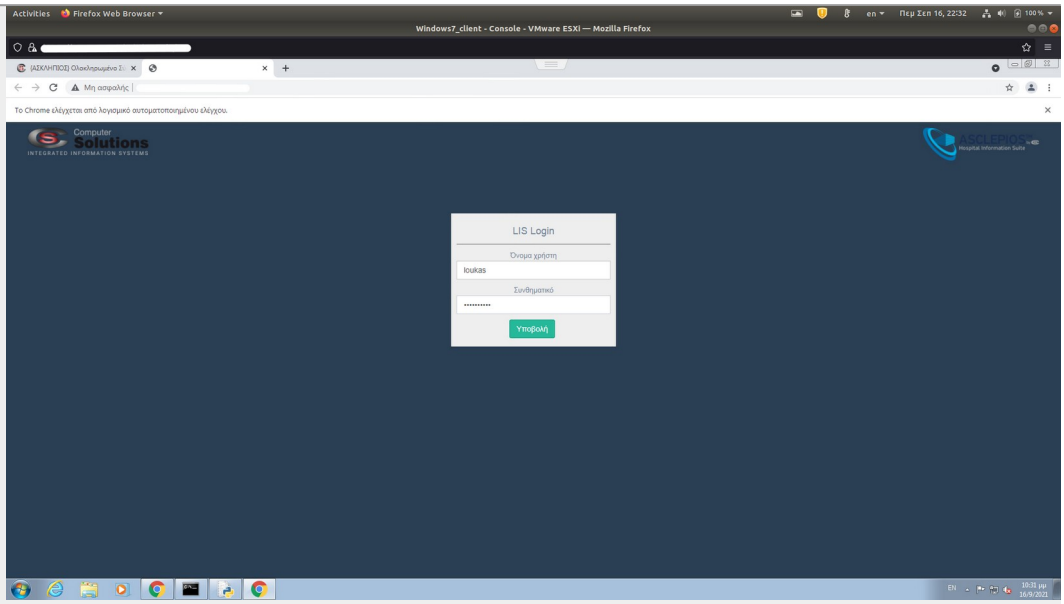


16 Επιλογή “Εκτύπωση παραπεμπτικών”.

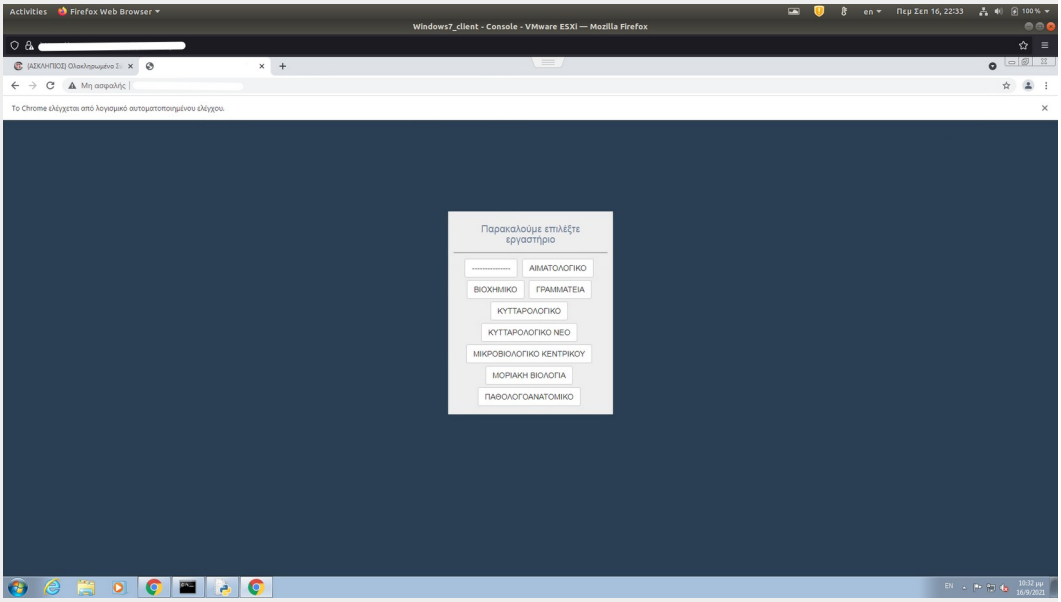


17 Εμφάνιση του παραπεμπτικού.

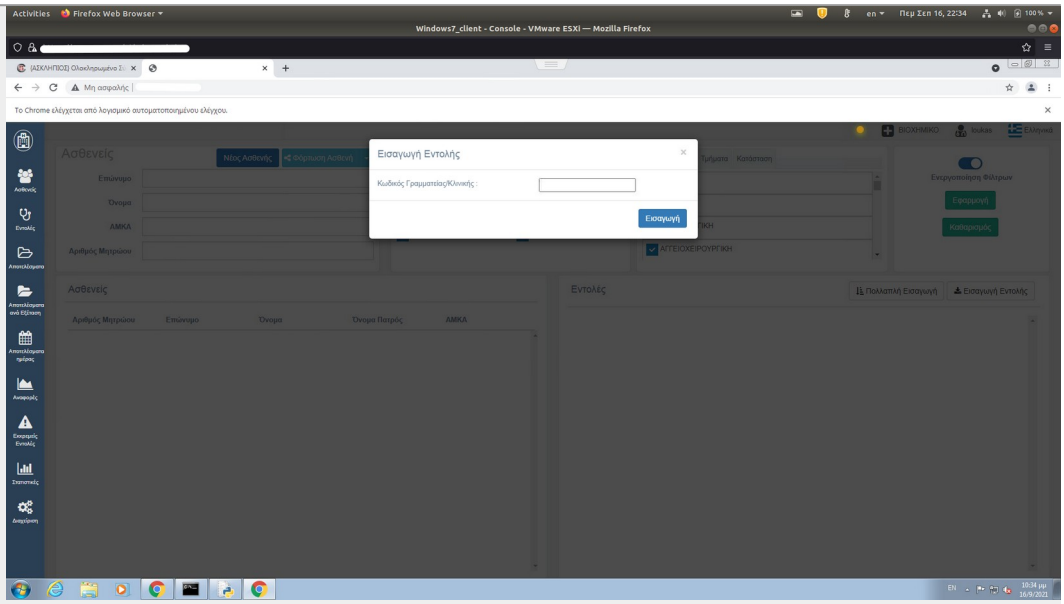
	
18	Κλείσιμο Ασκληπιού. **Αν η παράμετρος “asklipios_only” στο αρχείο LIS_config.json υπό το directory API/asklipios_LIS έχει την τιμή true , μετάβαση στο βήμα 1.**
19	Μετάβαση στο LIS.
20	Συμπλήρωση των πεδίων “Όνομα χρήστη” και “Συνθηματικό” και κατόπιν επιλογή “Υποβολή”. Σχόλιο: υπό το directory API/asklipios_LIS πρέπει να βρίσκεται ένα αρχείο .env, που θα περιλαμβάνει το username του χρήστη και τον κωδικό του χρήστη. Τα στοιχεία δεν χρειάζεται να είναι σε κάποια συγκεκριμένη σειρά, πέρα από το ότι κάθε ένα βρίσκεται σε νέα γραμμή εντός του αρχείου. Τα ίδια στοιχεία χρησιμοποιούνται και για την σύνδεση στον Ασκληπιό. **Επαφίεται στον χρήστη να εισάγει τα παραπάνω στοιχεία.** Παράδειγμα περιεχομένου του .env: USERNAME_LIS="XXXXXXXXX" PASSWORD_LIS="XXXXXXXXXX"



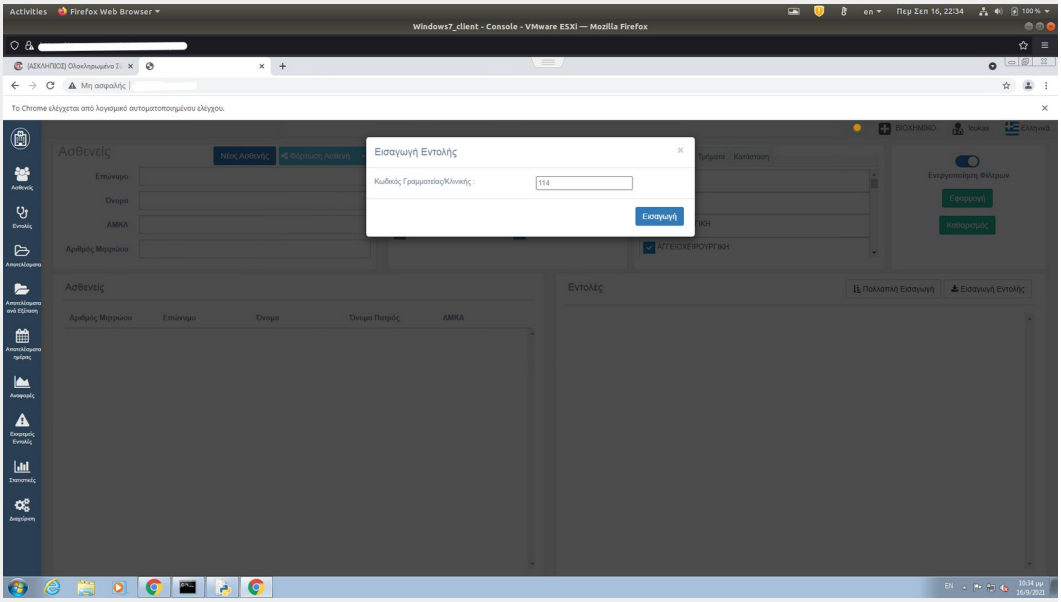
21 Επιλογή “ΒΙΟΧΗΜΙΚΟ”



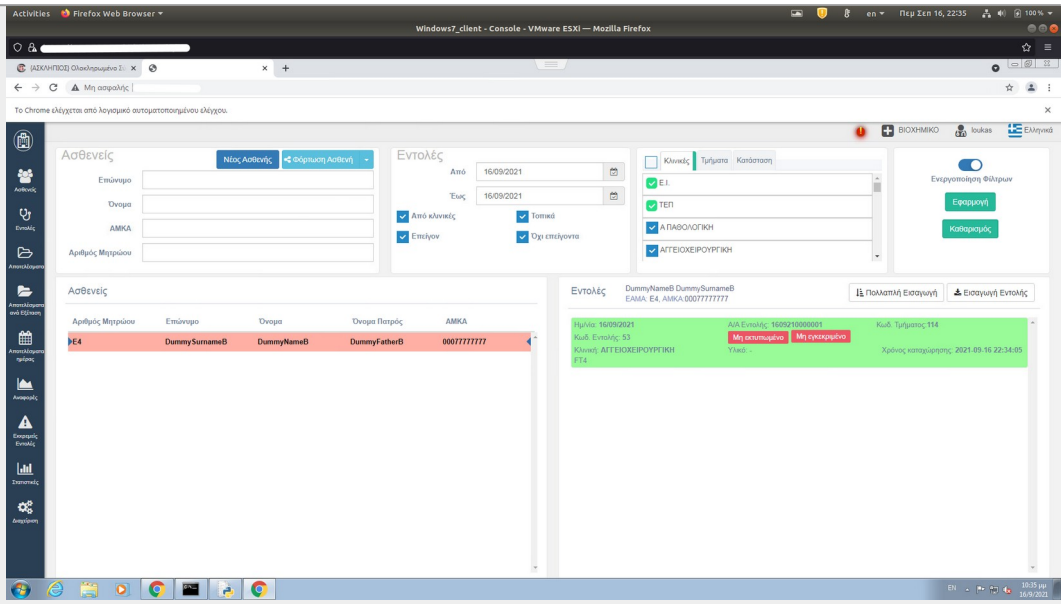
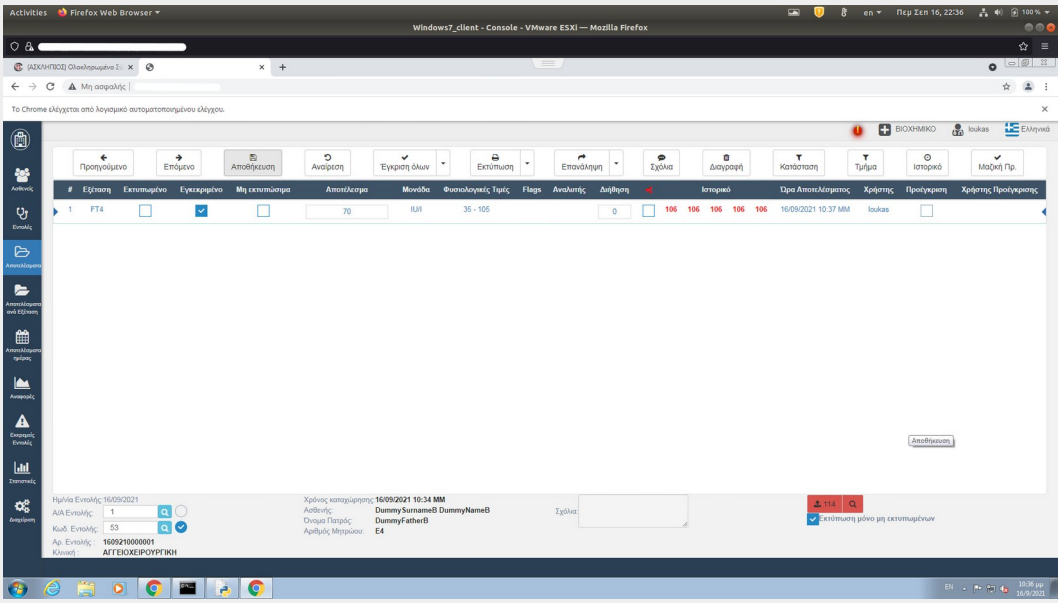
22 Επιλογή “Εισαγωγή Εντολής”.



23 Πληκτρολόγηση αριθμού εντολής.



24 Επιλογή με διπλό κλικ της τελευταίας εντολής.

	
25	Επιλογή “Εγκεκριμένο”.
	
26	Πιθανότητα να συμπληρωθεί το πεδίο των αποτελεσμάτων με τιμές μέσα ή έξω αντίστοιχα από τα φυσιολογικά όρια τιμών.
27	Επιλογή “Αποθήκευση”.
28	Επιλογή “Αποστολή Αποτελέσματος”.
29	Άνοιγμα του Ασκληπιού.

30	Έλεγχος για επιστροφή των αποτελεσμάτων στην στήλη “Κατάσταση” συνολικά για 10 λέπτα.
31	Κλείσιμο του Ασκληπιού.
32	Επανέναρξη του σεναρίου μέχρι το προφίλ να διακοπεί.

Πίνακας 4.2 Το σενάριο του Ασκληπιός – LIS

4.2.3.3 Persona

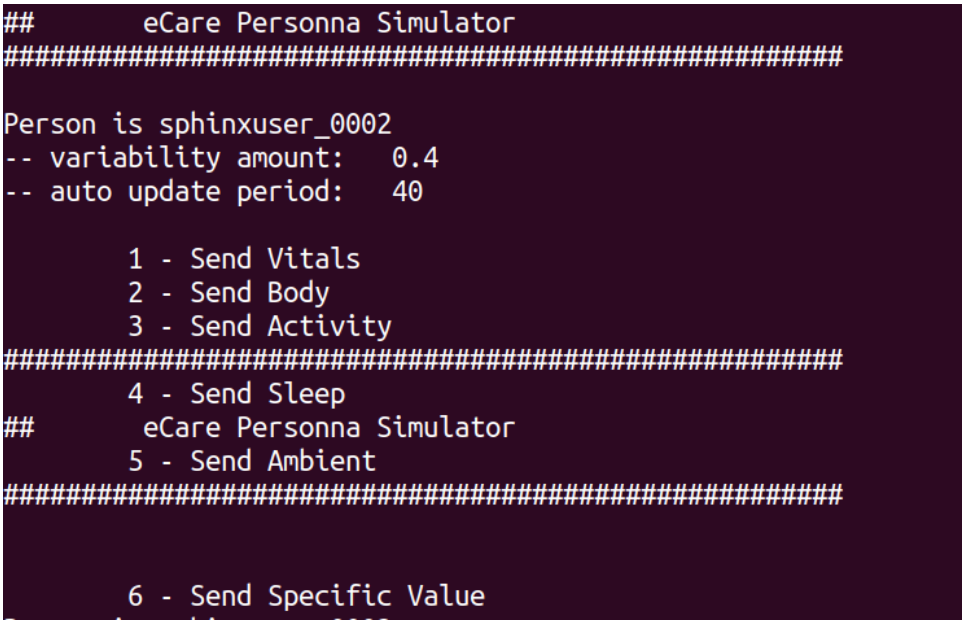
Η λειτουργικότητα της persona απλώς ενσωματώθηκε στο API, καθώς το λογισμικό της αναπτύχθηκε από τρίτους. Η persona δημιουργεί χρήστες (users) για τους οποίους αποστέλλει δεδομένα σε έναν server ανά τακτά χρονικά διαστήματα. Στη συνέχεια ο server απαντάει με μήνυμα επιτυχούς ή ανεπιτυχούς αντίστοιχα, λήψης των δεδομένων. Τα δεδομένα που αποστέλλονται είναι ιατρικής φύσεως, και αποτελούν χαρακτηριστικούς δείκτες της υγείας ενός ασθενούς όπως ο καρδιακός ρυθμός, η διαστολική και συστολική πίεση του αίματος κ.α. Για τους δείκτες αυτούς έχουν οριστεί κάποιες προκαθορισμένες τιμές και με βάση αυτές και την τιμή της διασποράς που έχει οριστεί για τον κάθε χρήστη, δημιουργούνται νέες τυχαίες τιμές.

Υπάρχει η δυνατότητα:

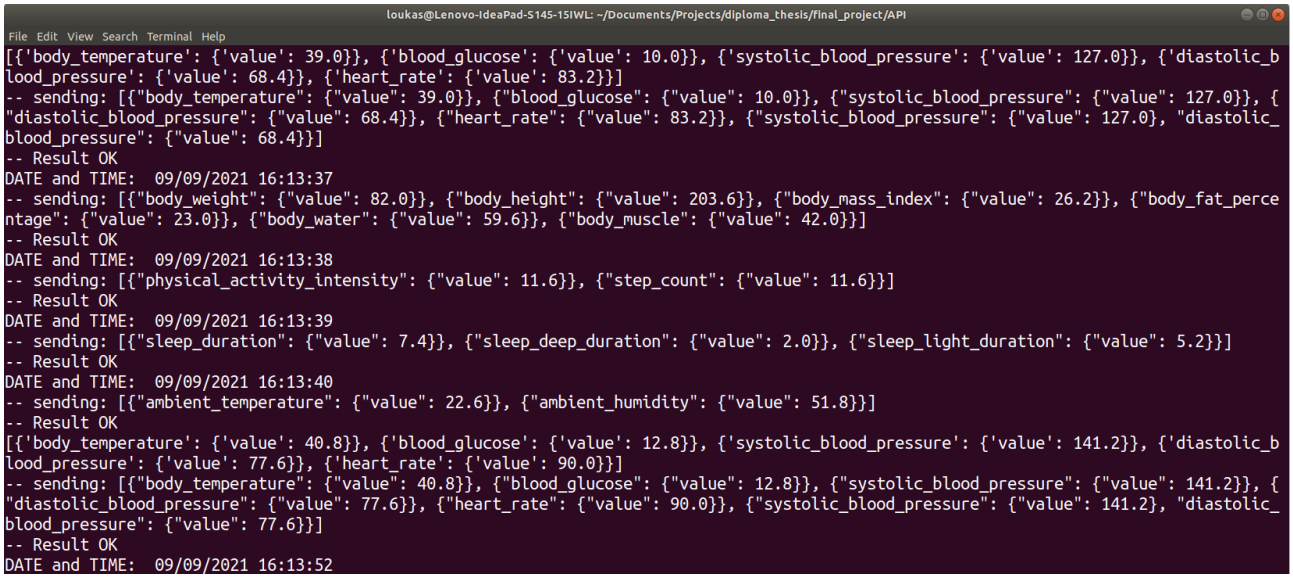
- Καθορισμού του πλήθους των ασθενών για τους οποίους θα υπάρχει ανταλλαγή πληροφορίων με τον server.

- Επιλογή της περιόδου ανά την οποία, θα γίνεται αίτημα αυτόματα από τον χρήστη και θα αποστέλλονται οι αντίστοιχες πληροφορίες από τον server.

- Επιλογή της διασποράς (variability), η οποία καθορίζει την απόκλιση από τις προκαθορισμένες τιμές για την δημιουργία νέων τυχαίων τιμών.



Εικόνα 4.4 Το μενού του UI της persona



Εικόνα 4.5 Αποστολή από τον client χαρακτηριστικών δεικτών υγείας του ασθενούς και απάντηση του server ότι τα δεδομένα ελήφθηκαν επιτυχώς

4.2.4 Βήματα για την εγκατάσταση του server

Στην πλευρά του server πρέπει να ακολουθηθεί η παρακάτω προεργασία:

i. Αποθήκευση των αρχείων του κώδικα σε κάποιο directory.

ii. Εγκατάσταση της python (έκδοση 3.6.9 ή νεώτερη)

iii. Εγκατάσταση των απαραίτητων python modules που περιέχονται στο αρχείο final_project/requirements.txt

   - Η εγκατάσταση σε περιβάλλον linux γίνεται μέσω της εντολής:

```
pip3 install -r requirements.txt
```

   - Ενώ η εγκατάσταση σε περιβάλλον windows γίνεται μέσω της εντολής:

```
pip install -r requirements.txt
```

iv. Εγκατάσταση των τελευταίων εκδόσεων των περιηγητών Firefox και Chrome.

v. Λήψη και αποθήκευση των αντίστοιχων των παραπάνω περιηγητών, chromedriver και geckodriver, σε κάποιο directory που να βρίσκεται στην enviromental μεταβλητή PATH (εναλλακτικά εισαγώγη του directory στο PATH)

vi. Εισαγωγή της επιθυμητής IP (και αντίστοιχου port) στην οποία θα «τρέχει» ο server, στην μεταβλητή host στο αρχείο final_project/server_config.json Για παράδειγμα:

```
“HOST” : “0.0.0.0:5000”
```

vii. Εκκίνηση του server με την εκτέλεση της εντολής:

   - Σε περιβάλλον linux

```
python3 server.py
```

   - Αντίστοιχα σε περιβάλλον windows

```
python server.py
```

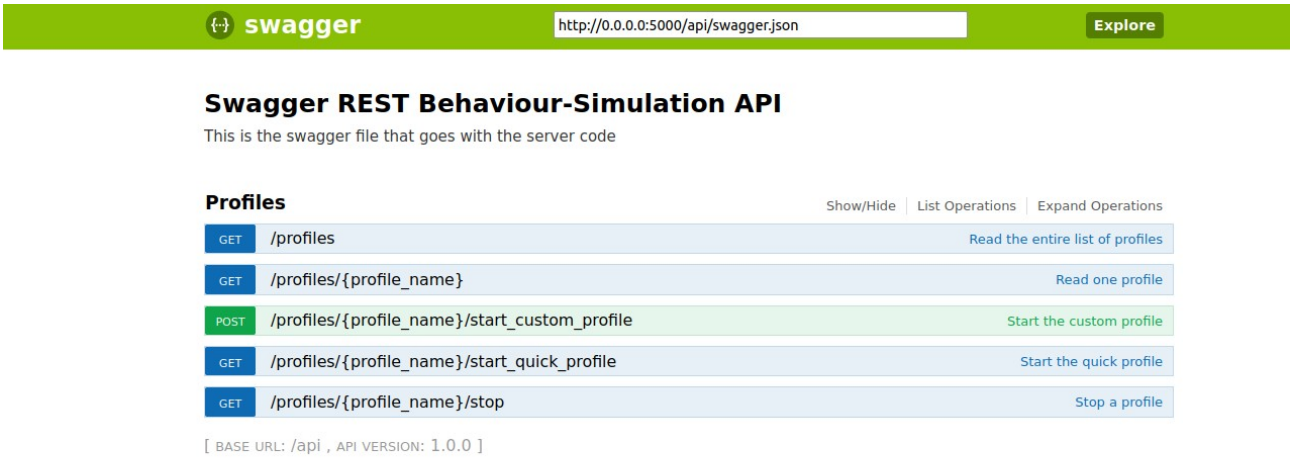
4.2.5 Διαχείριση των προφίλ μέσω του API

Η διαχείριση των προφίλ γίνεται από την πλευρά του client μέσω του διαθέσιμου API. Καθώς πρόκειται για Swagger API, τα HTTP requests μπορούν να γίνουν μέσω του UI (user interface) που παρέχει το Swagger.

Για να φτάσει ο client στο UI, αρκεί να «χτυπήσει» σε έναν browser το url: http://<ip:port>/api/ui/#/Profiles, όπου ip είναι η ip του server και port το αντίστοιχο port στο οποίο «ακούει» ο server, όπως έχουν οριστεί στην μεταβλητή host στο αρχείο server.py.

4.2.5.1 Αρχική σελίδα του UI

Η αρχική σελίδα του UI φαίνεται στο παρακάτω screenshot:



Εικόνα 4.6 Η αρχική σελίδα του Swagger UI

4.2.5.2 Ανάκτηση κατάστασης και πληροφοριών όλων των διαθέσιμων προφίλ

Μέσω του endpoint http://<ip:port>/api/profiles ο client μπορεί να ζητήσει πληροφορίες για την κατάσταση όλων των διαθέσιμων προφίλ. Αν το request εξυπηρετηθεί επιτυχώς ο server αποστέλλει τον κωδικό 200 ενώ στο response body υπάρχει η ζητούμενη πληροφορία.

Οι πληροφορίες που αποστέλλονται είναι οι παρακάτω:

profile name	Το όνομα του προφίλ
status	Αν το προφίλ έχει εκκινηθεί
duration	Η διάρκεια για την οποία το προφίλ θα «τρέχει»
started running	Timestamp της μέρας και ώρας εκκινήσεως

time remaining	Εναπομένων χρόνος
request body	Το body του request με το οποίο εκκινήθηκε το προφίλ (Αφορά το custom προφίλ μόνο)
applications	Οι εφαρμογές που εκκινήθηκαν
persona	Αν έχει εκκινηθεί το υποσύστημα persona
asklipios/LIS	Αν έχει εκκινηθεί το υποσύστημα Ασκληπιός/LIS

Πίνακας 4.3 Πληροφορίες για το εκάστοτε προφίλ που αποστέλλει ο server

Εικόνα 4.7α Παράδειγμα response body, στο οποίο και τα δύο διαθέσιμα προφίλ δεν έχουν εκκινηθεί

Request URL

```
http://0.0.0.0:5000/api/profiles
```

Response Body


```
        "outlook": {
          "status": "not running"
        },
        "promitheus": {
          "status": "not running"
        },
        "youtube": {
          "status": "not running"
        }
      },
      "asklipios_LIS": "not running",
      "duration": null,
      "persona": "not running",
      "profile_name": "Quick_Profile",
      "request_body": null,
      "started_running": null,
      "status": "not running",
      "time_remaining": null
    }
]
```


Response Code

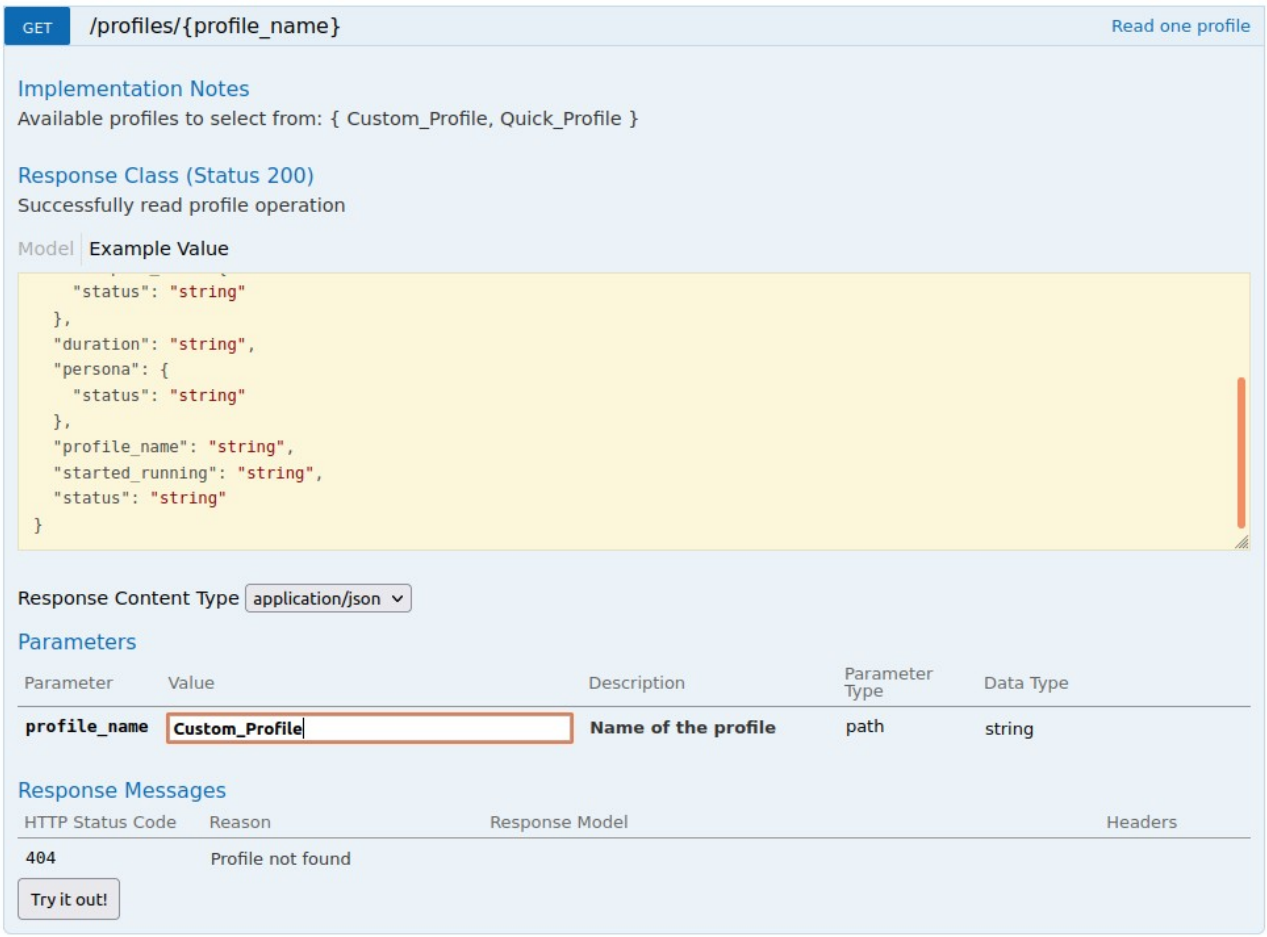
```
200
```

Εικόνα 4.7β Παράδειγμα response body, στο οποίο και τα δύο διαθέσιμα προφίλ δεν έχουν εκκινηθεί

4.2.5.3 Ανάκτηση κατάστασης και πληροφοριών ενός προφίλ

Μέσω του endpoint http://<ip:port>/api/profiles/{profile_name} ο client μπορεί να ζητήσει πληροφορίες για την κατάσταση ενός εκ των δύο διαθέσιμων προφίλ. Στην φόρμα profile_name συμπληρώνει το όνομα του αντίστοιχου προφίλ (Custom Profile ή Quick Profile). Αν το request εξυπηρετηθεί επιτυχώς ο server αποστέλλει τον κωδικό 200 ενώ στο response body υπάρχει η ζητούμενη πληροφορία. Αν το όνομα που δόθηκε δεν αντιστοιχεί σε κάποιο προφίλ αποστέλλεται ο κωδικός 404 με αντίστοιχο μήνυμα.

Οι πληροφορίες που αποστέλλονται φαίνονται στην παράγραφο 4.5.2



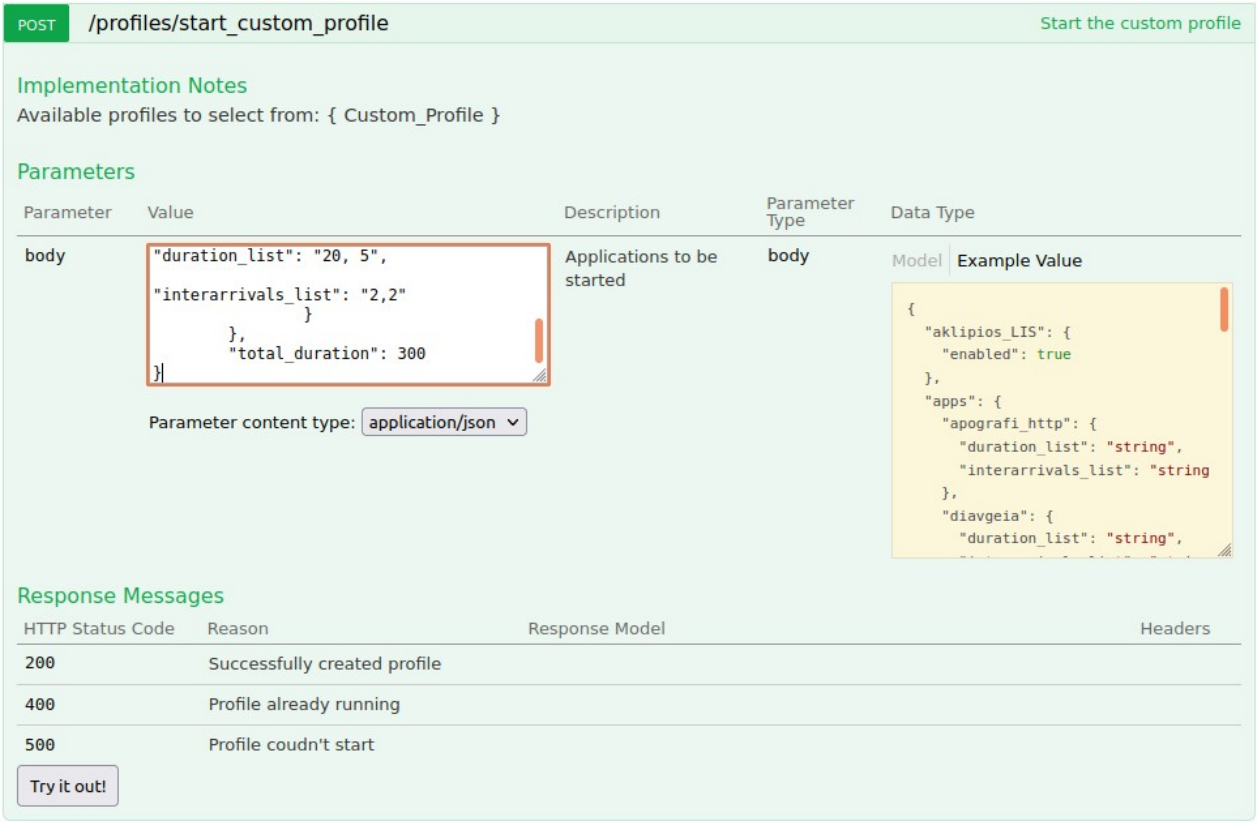
Εικόνα 4.8 Παράδειγμα, στο οποίο θα ζητηθεί πληροφορία για το Custom Profile

4.2.5.4 Εκκίνηση των προφίλ

4.2.5.4.1 Custom Profile

4.2.5.4.1.1 Εκκίνηση του Custom Profile

Μέσω του endpoint http://<ip:port>/api/profiles/start_custom_profile ο client μπορεί να εκκινήσει το Custom Profile. Επειδή πρόκειται για POST request, στην φόρμα body εισάγει τα επιθυμητά χαρακτηριστικά του προφίλ που θα εκκινηθεί, σε μορφή json. Αν το προφίλ εκκινηθεί επιτυχώς επιστρέφεται ο κωδικός 200 και ένα αντίστοιχο μήνυμα. Αν το προφίλ έχει ήδη εκκινηθεί επιστρέφεται ο κωδικός 400 και αντίστοιχο μήνυμα. Τέλος αν προκύψει οποιοδήποτε άλλο σφάλμα κατά την εκκίνηση του προφίλ επιστρέφεται ο κωδικός 500.



Εικόνα 4.9 Παράδειγμα εκκίνησης του Custom Profile

Εικόνα 4.10 Απάντηση του server με τον κωδικό 200 και αντίστοιχο μήνυμα κατόπιν επιτυχημένης εκκινήσεως του Custom Profile

4.2.5.4.1.2 Body του request

Όπως αναφέρθηκε και παραπάνω το body του POST request για την εκκίνηση του Custom Profile πρέπει να είναι σε μορφή json.

Όλα τα πεδία φαίνονται αναλυτικά στο παρακάτω json.

```
{
"asklipios_LIS": {
     "enabled": boolean
     },
  "persona": {
     "enabled": boolean
     },
  "apps" : {
             "google": {
                        "duration_list": "string",
                        "interarrivals_list": "string",
                       },
             "youtube": {
                        "duration_list": "string",
                        "interarrivals_list": "string",
                       },
             "facebook": {
                        "duration_list": "string",
                        "interarrivals_list": "string",
                       },
             "outlook": {
                        "duration_list": "string",
                        "interarrivals_list": "string",
                       },
             "promitheus": {
                        "duration_list": "string",
                        "interarrivals_list": "string",
                       },
             "galinos": {
                        "duration_list": "string",
                        "interarrivals_list": "string",
                       },
             "apografi_http": {
                        "duration_list": "string",
                        "interarrivals_list": "string",
                       },
             "gsis": {
                        "duration_list": "string",
                        "interarrivals_list": "string",
                       },
             "idika": {
                        "duration_list": "string",
                        "interarrivals_list": "string",
```

```
                    },
            "ebaby": {
                        "duration_list": "string",
                        "interarrivals_list": "string",
                    },
            "eopyy": {
                        "duration_list": "string",
                        "interarrivals_list": "string",
                    },
            "e_prescription": {
                        "duration_list": "string",
                        "interarrivals_list": "string",
                    },
            "diavgeia": {
                        "duration_list": "string",
                        "interarrivals_list": "string",
                    },
            "e_services": {
                        "duration_list": "string",
                        "interarrivals_list": "string",
                    },
            "dypethessaly": {
                        "duration_list": "string",
                        "interarrivals_list": "string",
                    }
    },
  "total_duration": int
  }
```

4.2.5.4.1.3 Συνολική χρονική διάρκεια του προφίλ

Στο πεδίο “total_duration” καθορίζεται ο συνολικός χρόνος κατά τον οποίο θα «τρέξει» το προφίλ. Με το πέρας του, το προφίλ τερματίζεται αυτόματα. Ο χρόνος πρέπει να δίνεται στην μονάδα των δευτερολέπτων και ως φυσικό συνεπακόλουθο πρέπει να είναι θετικός αριθμός.

4.2.5.4.1.4 Επιλογή Εφαρμογών

Για να επιλέξουμε κάποια από τις διαθέσιμες εφαρμογές, την ενθυλακώνουμε υπό το πεδίο “apps” και απαραίτητα πρέπει να συνοδεύεται από τα δύο πεδία της, “duration_list” και “interarrivals_list” που περιγράφονται παρακάτω. Το πεδίο “apps” πρέπει να είναι πάντα παρών αλλά μπορεί να μην περιέχει στοιχεία.

4.2.5.4.1.5 Duration List και Interarrivals List

Κάθε εφαρμογή πρέπει να συνοδεύεται από τα πεδία “duration_list” και “interarrivals_list” που πρέπει να είναι τύπου string. Τα δύο strings πρέπει να είναι ίδιου μεγέθους (να έχουν ίδιο αριθμό στοιχείων) ,καθώς υπάρχει αντιστοίχιση ένα προς ένα μεταξύ τους. Στο duration_list περιέχονται οι διάρκειες σε δευτερόλεπτα στις οποίες θα «τρέχει» η αντίστοιχη

εφαρμογή, ενώ στο interarrivals_list περιέχονται οι διάρκειες, επίσης σε δευτερόλεπτα, των ενδιάμεσων αντίστοιχων διαλειμμάτων. Οι τιμές διαχωρίζονται εντός του string με τον χαρακτήρα κόμμα (“,”).

4.2.5.4.1.6 Παράδειγμα Custom Profile

Για παράδειγμα για να «τρέξουν» οι εφαρμογές Facebook και Outlook, το json διαμορφώνεται ως εξής:

```
{
"asklipios_LIS": {
     "enabled": true
     },
"persona": {
     "enabled": false
     },
"apps" : {
     "facebook": {
               "duration_list": "10, 20, 60",
                "interarrivals_list": "3, 10, 5",
          },
     "outlook": {
               "duration_list": "12, 12",
                "interarrivals_list": "0, 2",
          },
     },
  "total_duration": 300
}
```

Το σενάριο της εφαρμογής facebook θα τρέξει για 10 δευτερόλεπτα, θα ακολουθήσει ένα διάστημα αδράνειας (sleep time) 3 δευτερολέπτων και θα ξαναρχίσει αυτή τη φορά να τρέχει για 20 δευτερόλεπτα κ.ο.κ. Αντίστοιχα και για το σενάριο της εφαρμογής outlook: θα τρέξει για 12 δευτερόλεπτα, δεν θα ακολουθήσει ένα διάστημα αδράνειας (sleep time) καθότι η πρώτη τιμή της interarrivals_list είναι μηδέν και θα ξαναρχίσει αυτή τη φορά να τρέχει για 12 δευτερόλεπτα κ.ο.κ.

Στο παραπάνω παράδειγμα θα εκκινηθεί και ο Ασκληπιός ενώ δεν θα ενεργοποιηθεί η persona. Σημειώνεται ότι στο πεδίο “apps” μπορεί να μην ενθυλακωθεί καμμία εφαρμογή αλλά σαν πεδίο πρέπει να υπάρχει αλλιώς θα επιστραφεί σφάλμα.

4.2.5.4.1.7 Παρατηρήσεις για την εκκίνηση του Custom Profile

Παρακάτω, παρουσιάζονται συνοπτικά κανόνες που πρέπει να ακολουθεί το body του POST request. Σφάλμα επιστρέφεται στις παρακάτω περιπτώσεις:

- Απουσιάζει κάποιο από τα παρακάτω υποχρεωτικά πεδία:

```
{
"asklipios_LIS": {
     "enabled": true
     },
"persona": {
     "enabled": false
     },
"apps" : {

     },
  "total_duration": 300
}
```

- Οι τιμές των πεδίων “enabled” έχουν κάποια άλλη τιμή πέραν των λογικών, true ή false.

- Η τιμή του πεδίου “total_duration” δεν είναι θετικός ακέραιος. Επισημαίνεται ότι η τιμή δίνεται σε δευτερόλεπτα.

- Οι λίστες “duration_list” και “interarrivals_list” δεν έχουν το ίδιο πλήθος στοιχείων.

4.2.5.4.1.8 Παραμετροποίηση του Ασκληπιού/LIS

Η ενεργοποίηση της παραπάνω λειτουργικότητας γίνεται μέσω του boolean πεδίου “enabled” υπό το πεδίο “asklipios_LIS”. Ο client δίνει είτε την τιμή true είτε την τιμή false. Η παραμετροποίηση του Ασκληπιού γίνεται μέσω του αρχείου LIS_config.json, οποίο βρίσκεται την εξής σχετική διεύθυνση: final_project/API/asklipios_LIS/

Το περιεχόμενο του παραπάνω αρχείου είναι το παρακάτω json:

```
{
     "LIS": "http://<LIS_IP>",
     "asklipios": "http://<asklipios_IP>/",
     "asklipios_only": true,
     "time_between_sessions": 10,
     "time_between_asklipios_LIS": 10,
     "epigon_probability": 0.5,
     "normal_values_probability": 0.5
}
```

- Στα πεδία “LIS” και “asklipios” συμπληρώνονται οι IP στις οποίες τρέχουν οι συγκεκριμένες εφαρμογές.

- Στο πεδίο “asklipios_only” συμπληρώνεται με boolean τιμή (true ή false) αν το σενάριο που θα τρέξει θα είναι μόνο αυτού του Ασκληπιού.

- Στο πεδίο “time_between_sessions” συμπληρώνεται ο χρόνος που επιθυμεί ο χρήστης να συμπληρωθεί μέχρι να επανεκκινηθεί το σενάριο. Η τιμή δίνεται σε δευτερόλεπτα.

- Στο πεδίο “time_between_asklipios_LIS” συμπληρώνεται ο χρόνος που επιθυμεί ο χρήστης να διαμεσολαβεί ανάμεσα στο σενάριο του Ασκληπιού και το σενάριο του LIS. Η τιμή δίνεται σε δευτερόλεπτα.

- Στο πεδίο “epigon_probability” συμπληρώνεται η πιθανότητα να καταχωρηθεί μία εξέταση ως επείγουσα.

- Στο πεδίο “normal_values_probability” συμπληρώνεται η πιθανότητα οι εξετάσεις του ασθενούς να έχουν τιμές εντός των φυσιολογικών οριών.

Το σενάριο του Ασκληπιού/LIS τρέχει συνεχώς μέχρι να τερματιστεί το προφίλ είτε μέσω του API από τον χρήστη, είτε μέσω του timeout timer που έχει συμπληρωθεί στο πεδίο “total_duration”.

4.2.5.4.1.9 Παραμετροποίηση της Persona

Η ενεργοποίηση της παραπάνω λειτουργικότητας γίνεται μέσω του boolean πεδίου “enabled” υπό το πεδίο “persona”. Ο client δίνει είτε την τιμή true είτε την τιμή false. Η παραμετροποίση της perona γίνεται μέσω του αρχείου persona_config.json, το οποίο βρίσκεται στην εξής σχετική διεύθυνση: final_project/API/persona/

Το περιεχόμενο του παραπάνω αρχείου είναι το παρακάτω json:

```
{
    "number_of_users": 3,
    "variability_list": [0.2, 0.4, 0.5],
    "period_list": [20, 40, 100]
}
```

- Στο πεδίο “number_of_users” συμπληρώνεται το πλήθος των χρηστών που θα δημιουργηθούν.

- Το “variability_list” είναι η λίστα με τις διασπορές που αντιστοιχεί σε κάθε χρήστη.

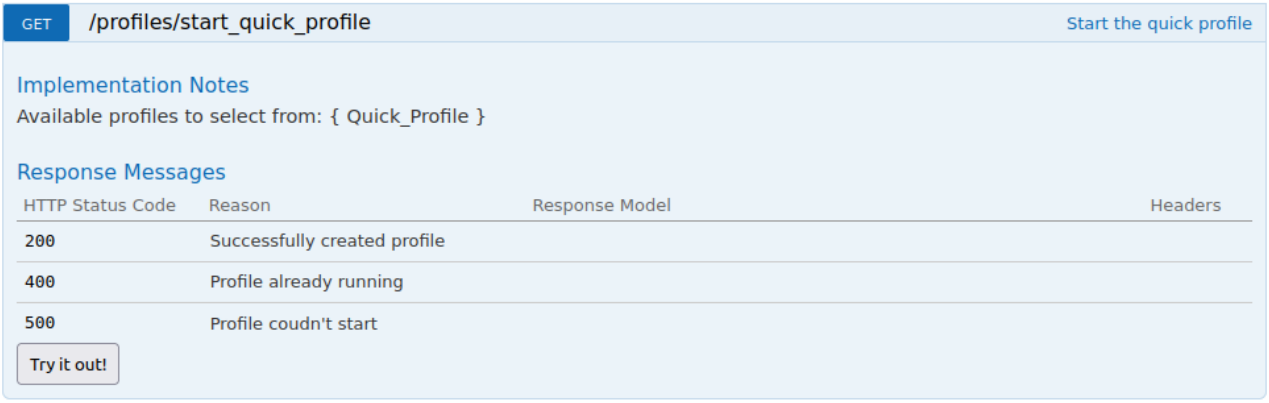
- Το “period_list” είναι η λίστα με τις περιόδους σε δευτερόλεπτα ανά τις οποίες θα επικοινωνεί ο κάθε χρήστης με τον νοσοκομειακό server.

4.2.5.4.2 Quick Profile

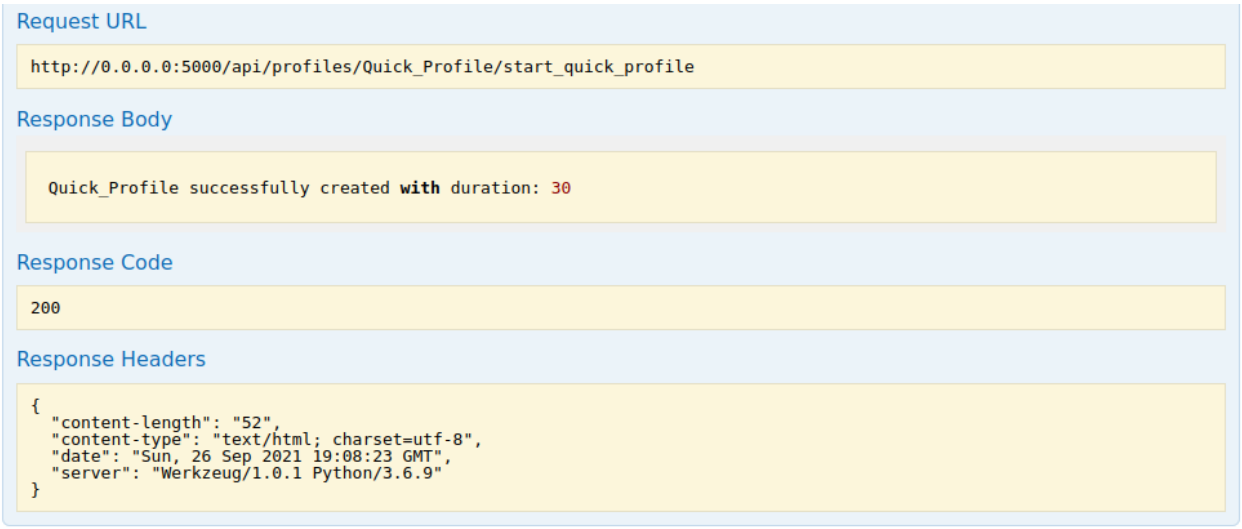
4.2.5.4.2.1 Εκκίνηση του Quick Profile

Μέσω του endpoint http://<ip:port>/api/profiles/{profile_name}/start_quick_profile ο client μπορεί να εκκινήσει το Quick Profile. Αν το προφίλ εκκινηθεί επιτυχώς επιστρέφεται ο κωδικός 200 και ένα αντίστοιχο μήνυμα. Αν το προφίλ έχει ήδη εκκινηθεί επιστρέφεται ο

κωδικός 400 και αντίστοιχο μήνυμα. Τέλος αν προκύψει οποιοδήποτε άλλο σφάλμα κατά την εκκίνηση του προφίλ επιστρέφεται ο κωδικός 500.



Εικόνα 4.11 Παράδειγμα εκκινήσεως του Quick Profile



Εικόνα 4.12 Απάντηση του server με τον κωδικό 200 και αντίστοιχο μήνυμα κατόπιν επιτυχημένης εκκινήσεως του Quick Profile

Εικόνα 4.13 Απάντηση του server με τον κωδικό 400 και αντίστοιχο μήνυμα κατόπιν αποτυχημένης εκκινήσεως του Quick Profile

4.2.5.4.2.2 Παραμετροποίηση του Quick Profile

Η παραμετροποίηση του Quick Profile γίνεται από την πλευρά του server μέσα από το αρχείο quick_profile_config.json το οποίο βρίσκεται στην εξής σχετική διεύθυνση: final_project/API/

Το περιεχόμενο του παραπάνω αρχείου είναι το παρακάτω json:

```
{
   "COMMENTS" : "ATTENTION: to not use an app its sessions
must be             0. For asklipios and LIS toggle enabled
between                  true/false",

  "asklipios_LIS": {
     "enabled": false
     },
  "persona": {
     "enabled": false
     },
  "apps" : {
            "google": {
                        "total_sessions_duration": 10,
                        "total_interarrivals_duration": 5,
                        "sessions": 2
                      },
            "youtube": {
                        "total_sessions_duration": 10,
                        "total_interarrivals_duration": 10,
                        "sessions": 0
                      },
            "facebook": {
                        "total_sessions_duration": 10,
                        "total_interarrivals_duration": 10,
```

```
            "sessions": 0
        },
    "outlook": {
            "total_sessions_duration": 10,
            "total_interarrivals_duration": 10,
            "sessions": 0
        },
    "promitheus": {
            "total_sessions_duration": 100,
            "total_interarrivals_duration": 10,
            "sessions": 0
        },
    "galinos": {
            "total_sessions_duration": 100,
            "total_interarrivals_duration": 10,
            "sessions": 0
        },
    "apografi_http": {
            "total_sessions_duration": 10,
            "total_interarrivals_duration": 10,
            "sessions": 0
        },
    "gsis": {
            "total_sessions_duration": 10,
            "total_interarrivals_duration": 10,
            "sessions": 0
        },
    "idika": {
            "total_sessions_duration": 10,
            "total_interarrivals_duration": 10,
            "sessions": 0
        },
    "ebaby": {
            "total_sessions_duration": 10,
            "total_interarrivals_duration": 10,
            "sessions": 0
        },
    "eopyy": {
            "total_sessions_duration": 10,
            "total_interarrivals_duration": 10,
            "sessions": 0
        },
    "e_prescription": {
            "total_sessions_duration": 10,
            "total_interarrivals_duration": 10,
            "sessions": 0
        },
    "diavgeia": {
            "total_sessions_duration": 10,
            "total_interarrivals_duration": 10,
            "sessions": 0
```

```
                    },
            "e_services": {
                        "total_sessions_duration": 10,
                        "total_interarrivals_duration": 10,
                        "sessions": 0
                    },
            "dypethessaly": {
                        "total_sessions_duration": 0,
                        "total_interarrivals_duration": 10,
                        "sessions": 0
                    }
        },
  "total_duration": 30
  }
```

4.2.5.4.2.3 Ανάλυση των πεδίων του αρχείου

4.2.5.4.2.3.1 Total_duration

Στο πεδίο “total_duration” καθορίζεται ο συνολικός χρόνος κατά τον οποίο θα «τρέξει» το προφίλ. Με το πέρας του, το προφίλ τερματίζεται αυτόματα. Ο χρόνος πρέπει να δίνεται στην μονάδα των δευτερολέπτων και ως φυσικό συνεπακόλουθο πρέπει να είναι θετικός αριθμός.

4.2.5.4.2.3.2 Sessions

Στο πεδίο “sessions” καθορίζεται πόσες φορές θα επανεκκινηθεί, η συγκεκριμένη εφαρμογή υπό την οποία έχει ενθυλακωθεί το πεδίο. Σε κάθε “session” αντιστοιχεί ένα χρονικό διάστημα στο οποίο η εφαρμογή θα «τρέχει», και ένα χρονικό διάστημα που έπεται του πρώτου, και η εφαρμογή παραμένει αδρανής (βλέπε Total_sessions_duration και Total_interarrivals_duration). Το πεδίο συμπληρώνεται με θετικό ακέραιο αριθμό. Εάν ο χρήστης δεν επιθυμεί την ενσώματωση στο προφίλ μια εφαρμογής, δίνει στο πεδίο την τιμή 0.

4.2.5.4.2.3.3 Total_sessions_duration

Στο πεδίο “total_sessions_duration” καθορίζεται ο συνολικός χρόνος που θα τρέξει η εφαρμογή υπό την οποία έχει ενθυλακωθεί το πεδίο. Κατόπιν, ο χρόνος αυτός κατανέμεται τυχαία στα αντίστοιχα sessions.

4.2.5.4.2.3.4 Total_interarrivals_duration

Στο πεδίο “total_interarrivals_duration” καθορίζεται ο συνολικός χρόνος των διαλειμμάτων ανάμεσα στις επανεκκινήσεις της εφαρμογής. Κατόπιν ο χρόνος αυτός κατανέμεται τυχαία στα αντίστοιχα διαλείμματα.

4.2.5.4.2.4 Ενεργοποίηση του Ασκληπιού - LIS

Η ενεργοποίηση της λειτουργικότητας του Ασκληπιού/LIS γίνεται μέσω του boolean πεδίου “enabled” υπό το πεδίο “asklipios_LIS”. Ο χρήστης του δίνει είτε την τιμή true είτε την τιμή false. Η παραμετροποίηση του Ασκληπιού γίνεται μέσω του αρχείου LIS_config.json, οποίο βρίσκεται την εξής σχετική διεύθυνση: final_project/API/asklipios_LIS/

Περισσότερες πληροφορίες για την παραμετροποίηση στην παράγραφο 5.4.1.8

4.2.5.4.2.5 Ενεργοποίηση της persona

Η ενεργοποίηση της παραπάνω λειτουργικότητας γίνεται μέσω του boolean πεδίου “enabled” υπό το πεδίο “persona”. Ο χρήστης του δίνει είτε την τιμή true είτε την τιμή false. Η παραμετροποίηση της persona γίνεται μέσω του αρχείου persona_config.json, το οποίο βρίσκεται στην εξής σχετική διεύθυνση: final_project/API/persona/

Περισσότερες πληροφορίες για την παραμετροποίηση στην παράγραφο 5.4.1.9

4.2.5.4.2.6 Για την ενεργοποίηση των εφαρμογών

Για να ενεργοποιηθεί μία εφαρμογή πρέπει η τιμή του πεδίου “sessions” υπό το όνομα της εφαρμογής να είναι 1 ή μεγαλυτερος θετικός ακέραιος αριθμός, ανάλογα με το πόσες φορές επιθυμεί ο χρήστης να επαναληφθεί το σενάριο. Για να είναι ανενεργή μία εφαρμογή πρέπει το αντίστοιχο πεδίο “sessions” να έχει την τιμή 0.

4.2.5.4.2.7 Παράδειγμα Quick Profile 1

```
"facebook": {
"total_sessions_duration": 10,
"total_interarrivals_duration": 10,
"sessions": 3
                    }
```

Στο παραπάνω παράδειγμα η εφαρμογή facebook θα «τρέξει» 3 φορές.
Πιθανή κατανομή των χρόνων παρουσιάζεται παρακάτω:

Session	Χρονική Διάρκεια Session (seconds)	Επακόλουθο Διάστημα Αδράνειας (seconds)
Πρώτο	2	3.5
Δεύτερο	3.4	2.3
Τρίτο	4.6	4.2
Άθροισμα:	10	10

Πίνακας 4.4 Πιθανή κατανομή χρόνων στις sessions

4.2.5.4.2.8 Παράδειγμα Quick Profile 2

```
"galinos": {
"total_sessions_duration": 100,
"total_interarrivals_duration": 10,
"sessions": 0
                        }
```

Στο παραπάνω παράδειγμα η εφαρμογή Γαληνός δεν θα «τρέξει» καθόλου.

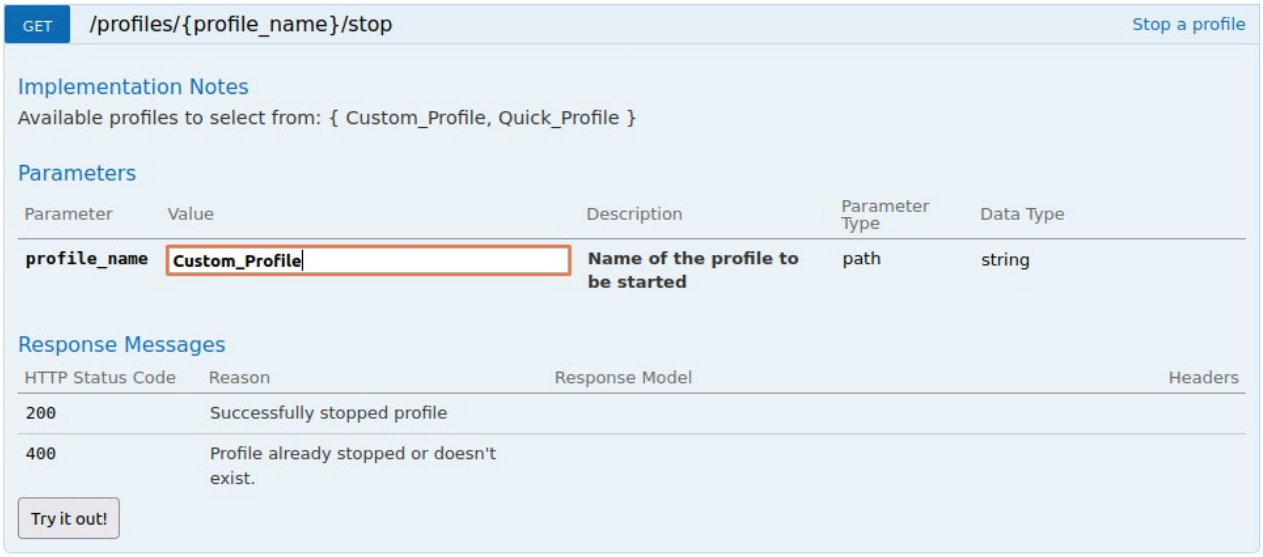
4.2.5.4.2.9 Παράδειγμα Quick Profile 3

```
"youtube": {
"total_sessions_duration": 10,
"total_interarrivals_duration": 0,
"sessions": 1
                        }
```

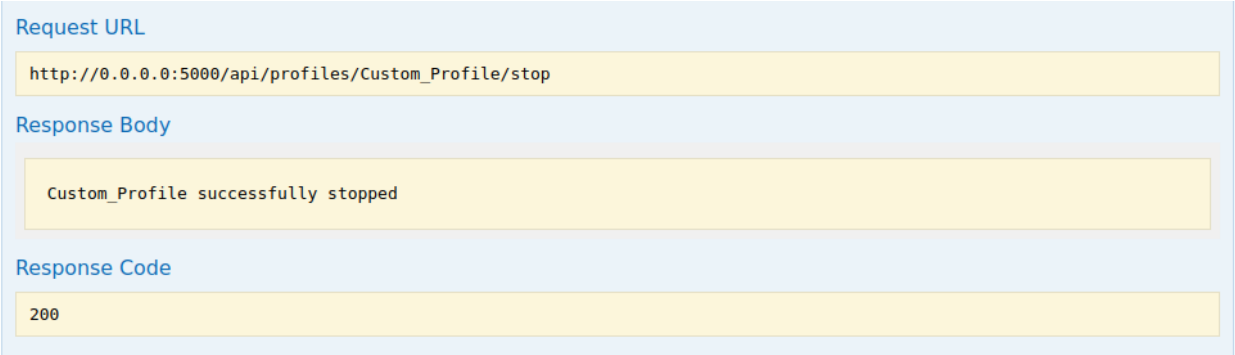
Στο παραπάνω παράδειγμα η εφαρμογή youtube θα «τρέξει» μία φορά με διάρκεια 10 δευτερολέπτων και δεν θα επακολουθήσει κάποιο διάλλειμα.

4.2.5.5 Τερματισμός ενός προφίλ

Μέσω του endpoint http://<ip:port>/api/profiles/{profile_name}/stop ο χρήστης μπορεί να τερματίσει το προφίλ του οποίου το όνομα έχει περαστεί στο path μέσω της παραμέτρου {profile_name}. Στο UI του API στο πεδίο profile_name συμπληρώνει είτε Custom_Profile είτε Quick_Profile. Αν το προφίλ τερματιστεί επιτυχώς επιστρέφεται ο κωδικός 200 και ένα αντίστοιχο μήνυμα. Αν το προφίλ έχει ήδη τερματιστεί ή δεν υπάρχει, επιστρέφεται ο κωδικός 400 και αντίστοιχο μήνυμα.



Εικόνα 4.14 Παράδειγμα τερματισμού του Custom Profile



Εικόνα 4.15 Απάντηση του server με τον κωδικό 200 και αντίστοιχο μήνυμα κατόπιν επιτυχημένου τερματισμού του Custom Profile

Εικόνα 4.16 Απάντηση του server με τον κωδικό 400 και αντίστοιχο μήνυμα κατόπιν αποτυχημένου τερματισμού του Custom Profile

4.2.6 Διαστήματα αδράνειας μεταξύ ενεργειών

Πέραν των διαστημάτων αδράνειας ενδιάμεσα στην επανεκκίνηση των σεναρίων των προφίλ, στα οποία έγινε αναφορά παραπάνω, έχουν ενσωματωθεί και διαστήματα αδράνειας στα παρακάτω σημεία:

- Ανάμεσα σε ενέργειες που στην πραγματικότητα θα έκανε ο άνθρωπος-χρήστης , όπως για παράδειγμα την συμπλήρωση μιας φόρμας με το username του για να κάνει login, και την επόμενη ενέργεια, που ίσως είναι το click στην επιλογή submit κλπ. Τα διαστήματα αυτά αδράνειας έχουν ενσωματωθεί για να είναι όσο πιο ρεαλιστική γίνεται η προσομοίωση και προκύπτουν από μια γεννήτρια τυχαίων αριθμών x τέτοιων ώστε $\{1 \leq x \leq 4 \mid x \in \mathbb{Q}\}$.

- Ανάμεσα στην τελευταία ενέργεια του σενάριου του Ασκληπιού και την έναρξη του σεναρίου του LIS. Το διάστημα αυτό εκφράζει το χρόνο που διαμεσολαβεί ανάμεσα στην έναρξη των δύο λειτουργιών, σε μια πραγματική νοσοκομειακή εγκατάσταση.

4.2.7 Τεχνική περιγραφή των endpoints του Behaviour – Simulation API

Οι παρακάτω πίνακες περιγράφουν τα endpoints του API.

Endpoint Name	Ανάκτηση κατάστασης και πληροφοριών όλων των διαθέσιμων προφίλ
Description	Η σελίδα περιέχει πληροφορίες για τα δύο διαθέσιμα προφίλ (Custom Profile και Quick Profile). Οι πληροφορίες παρέχονται σε μορφή json. Οι διαθέσιμες πληροφορίες είναι τα παρακάτω πεδία: • profile name • status • duration • started running • time remaining • request body • applications • persona • asklipios/LIS
HTTP Method	GET
Endpoint URL	http://0.0.0.0:5000/api/profiles
Parameters	-
Requests Body	-

Πίνακας 4.5 Περιγραφή του endpoint http://0.0.0.0:5000/api/profiles

Endpoint Name	Ανάκτηση κατάστασης και πληροφοριών ενός προφίλ
Description	Η σελίδα περιέχει πληροφορίες για το προφίλ που έχει περαστεί σαν παράμετρος στο path. Οι πληροφορίες παρέχονται σε μορφή json. Οι διαθέσιμες πληροφορίες είναι τα παρακάτω πεδία:

	• profile name • status • duration • started running • time remaining • request body • applications • persona • asklipios/LIS
HTTP Method	GET
Endpoint URL	http://0.0.0.0:5000/api/profiles/{profile_name}

Parameters	{
	name: profile_name
	in: path
	type: string
	description: Name of the profile
	required: true
	}

Requests Body	-

Πίνακας 4.6 Περιγραφή του endpoint http://0.0.0.0:5000/api/profiles/{profile_name}

Endpoint Name	Έναρξη του Custom Profile
Description	Εκκινεί το Custom Profile. Οι απαιτούμενες πληροφορίες για το ποια application θα εκκινηθούν αποστέλλονται μέσω του body του POST request. Το όνομα του profile αποστέλλεται ως παράμετρος στο path του url και δεν μπορεί να απουσιάζει.
HTTP Method	POST
Endpoint URL	http://0.0.0.0:5000/api/profiles/start_custom_profile

Parameters	{
	name: body
	in: body
	description: Applications to be started
	}

Requests Body	{
	"asklipios_LIS": {
	"enabled": boolean
	},
	"persona": {
	"enabled": boolean
	},
	"apps" : {
	"google": {
	"duration_list": "string",
	"interarrivals_list": "string",
	},
	"youtube": {

```
            "duration_list": "string",
            "interarrivals_list": "string",
        },
    "facebook": {
            "duration_list": "string",
            "interarrivals_list": "string",
        },
    "outlook": {
            "duration_list": "string",
            "interarrivals_list": "string",
        },
    "promitheus": {
            "duration_list": "string",
            "interarrivals_list": "string",
        },
    "galinos": {
            "duration_list": "string",
            "interarrivals_list": "string",
        },
    "apografi_http": {
            "duration_list": "string",
            "interarrivals_list": "string",
        },
    "gsis": {
            "duration_list": "string",
            "interarrivals_list": "string",
        },
    "idika": {
            "duration_list": "string",
            "interarrivals_list": "string",
        },
    "ebaby": {
            "duration_list": "string",
            "interarrivals_list": "string",
        },
    "eopyy": {
            "duration_list": "string",
            "interarrivals_list": "string",
        },
    "e_prescription": {
            "duration_list": "string",
            "interarrivals_list": "string",
        },
    "diavgeia": {
            "duration_list": "string",
            "interarrivals_list": "string",
        },
    "e_services": {
            "duration_list": "string",
            "interarrivals_list": "string",
```

},
"dypethessaly": {
"duration_list": "string",
"interarrivals_list": "string",
}
},
"total_duration": int
}

Πίνακας 4.7 Περιγραφή του endpoint http://0.0.0.0:5000/api/profiles/start_custom_profile

Endpoint Name	Έναρξη του Quick Profile
Description	Εκκινεί το Quick Profile. Το configuration του προφίλ είναι αποθηκευμένο σε ένα config αρχείο στην πλευρά του server.
HTTP Method	GET
Endpoint URL	http://0.0.0.0:5000/api/profiles/start_quick_profile
Parameters	-
Requests Body	-

Πίνακας 4.8 Περιγραφή του endpoint http://0.0.0.0:5000/api/profiles/start_quick_profile

Endpoint Name	Τερματισμός ενός προφίλ
Description	Τερματίζει το προφίλ το όνομα του οποίου έχει περαστεί ως παράμετρος στο path (είτε Custom Profile είτε Quick Profile).
HTTP Method	GET
Endpoint URL	http://0.0.0.0:5000/api/profiles/{profile_name}/stop

Parameters	{
	name: profile_name
	in: path
	type: string
	description: Name of the profile
	required: true
	}

Requests Body	-

Πίνακας 4.9 Περιγραφή του endpoint http://0.0.0.0:5000/api/profiles/{profile_name}/stop

Κεφάλαιο 5

Επίλογος

5.1 Σύνοψη

Σκοπός της παρούσας διπλωματικής εργασίας ήταν η δημιουργία προσομοιωτή δικτυακής κίνησης σε νοσοκομειακές εγκαταστάσεις με απώτερο σκοπό την παραγωγή σετ δεδομένων (dataset) καλόβουλης δικτυακής κίνησης. Τα παραγώμενα σετ δεδομένων μετά και από τον εμπλουτισμό τους από κακόβουλη κίνηση, μπορούν να αξιοποιηθούν για την εκπαίδευση συστημάτων IDS. Η δυνατότητα αυτή που παρέχουν, αποκτά ιδιαίτερη αξία αν συνυπολογίσουμε ότι τα σετ δεδομένων που προέρχονται από αληθινές νοσοκομειακές εγκαταστάσεις, είτε δεν μπορούν να αξιοποιηθούν καθόλου, είτε κρίνονται ακατάλληλα καθώς έχουν επεξεργαστεί και χάσει σημαντικό κομμάτι πληροφορίας, για την προστασία των προσωπικών δεδομένων του προσωπικού και των ασθενών.

Αρχικά έγινε παρουσίαση του θεωρητικού υπόβαθρου για την ανάπτυξη του εργαλείου, που περιλάμβανε σύντομες παρουσιάσεις των IPS και IDS συστημάτων, των ήδη διαθέσιμων σετ δεδομένων και οι απαιτήσεις για ένα σύχρονο σετ, ανάλυση των εννοιών προφίλ και flow και σύντομη παρουσίαση των τεχνολογιών RESTful API και Swagger. Ακολούθησαν δυο προσεγγίσεις για την μοντελοποίηση της δικτυακής κίνησης, η χρήση ιστοσελίδων με στατιστικά για την πλοήγηση χρηστών στο διαδίκτυο και η στατιστική ανάλυση πραγματικής δικτυακής κίνησης όπου και αναλύθηκαν flows από νοσοκομειακό συγκρότημα. Στη συνέχεια αναλύθηκαν τα εργαλεία και οι τεχνολογίες που αξιοποιήθηκαν για την ανάπτυξη του προσομοιωτή και τέλος παρουσιάστηκε η ίδια η εφαρμογή. Η τελευταία επιτρέπει την δημιουργία προφίλ παραμετροποιήσιμων ως προς την διαδικτυακή συμπεριφορά τους, ώστε να μπορούν να προσομοιωθούν ικανοποιητικά ρόλοι όπως του γιατρού, του διοικητικού προσωπικού κλπ. Περιγράφηκε αναλυτικά ο έλεγχος της μέσω του Swagger RESTful API που αναπτύχθηκε, όπως και οι λειτουργικότητες της οι οποίες περιλαμβάνουν την εκκίνηση μιας σειράς από «εφαρμογές», στις οποίες γίνεται προσομοίωση της επικοινωνίας του προφίλ με τον client του εκάστοτε service (πχ Facebook, Youtube, Outlook κ.α.) και την προσομοίωση των ενδονοσοκομειακών συστημάτων Persona και Ασκληπιός – LIS.

5.2 Μελλοντικές Επεκτάσεις

Ο προσομοιωτής κρίνεται ότι ανταποκρίνεται ικανοποιητικά στον στόχο του, την παραγωγή δικτυακής κίνησης σε νοσοκομειακές εγκαταστάσεις. Η κίνηση αυτή θα μπορούσε να εμπλουτιστεί και με πακέτα πρωτόκολλων SKYPE, SSH και FTP, τεχνολογίες που εξερευνήθηκαν αλλά δεν ενσωματώθηκαν στην τελική εφαρμογή, ενώ εύκολη είναι και η ενσωμάτωση δικτυακής κίνησης που προέρχεται από περιήγηση σε ιστοσελίδες με κακόβουλο περιεχόμενο. Επίσης, επειδή το API αξιοποιεί την τεχνολογία Swagger, εύκολα μπορούν να ενσωματωθούν και άλλα προφίλ ανάλογα και με τις ανάγκες των προσομοιώσεων. Τέλος σαν επέκταση θα μπορούσε να θεωρηθεί η δημιουργία μίας client εφαρμογής που θα λειτουργεί ως κοινό σημείο ελέγχου των εκάστοτε προφίλ του Behaviour Simulator σε ένα δίκτυο υπολογιστών, παρέχοντας και ένα φιλικό προς τον χρήστη GUI που θα απλουστεύσει την διαχείριση των προφίλ.

Παράρτημα Κώδικα

Κ.1 Ενδεικτικός κώδικας με τον οποίο αυτοματοποιείται το online browser παιχνίδι cookie clicker[27]:

```
from selenium import webdriver
from selenium.webdriver.common.action_chains import
ActionChains

PATH = "/home/loukas/chromedriver"
driver = webdriver.Chrome(PATH)
driver.get("https://orteil.dashnet.org/
cookieclicker")

driver.implicitly_wait(5)

cookie = driver.find_element_by_id("bigCookie")
cookie_count = driver.find_element_by_id("cookies")
items = [driver.find_element_by_id("productPrice" +
str(i)) for i in range(1, -1, -1)]

actions = ActionChains(driver)
actions.click(cookie)

for i in range(5000):
     actions.perform()
     count = int(cookie_count.text.split(" ")[0])
     for item in items:
          value = int(item.text)
          if value <= count:
               upgrade_actions = ActionChains(driver)
               upgrade_actions.move_to_element(item)
               upgrade_actions.click()
               upgrade_actions.perform()
```

27 [https://orteil.dashnet.org/cookieclicker/]

Κ.2 Ενδεικτικός κώδικας με τον οποίο αυτοματοποιείται η εφαρμογή Skype

```
from skpy import Skype, SkypeChats
from dotenv import load_dotenv
import os, random, string, sys

def main():

     # Sign in.
     try:
          load_dotenv()
          password = os.environ.get('SKYPE_PASSWORD')
          email = os.environ.get('SKYPE_EMAIL')
          sk = Skype(email, password)
     except:
          print("sign in error: " + str(e))
          sys.exit()

     # Get a random contact.
     try:
          chats = list(SkypeChats(sk).recent().keys())
          chat = sk.chats[random.choice(chats)]
     except:
          print("get contact error: " + str(e))
          sys.exit()

     # Read old messages
     try:
          chat.getMsgs()
     except:
          print("read message error: " + str(e))
          sys.exit()

     # Create a random message.
     letters = string.ascii_letters
     random_string = ''.join(random.choice(letters) for i
in range(10))

     # Send message.
     try:
          msg = chat.sendMsg(random_string)
     except:
          print("send message error: " + str(e))
          sys.exit()

if __name__ == "__main__":
     main()
```

Κ.3 Ενδεικτικός κώδικας με τον οποίο αυτοματοποιείται η σύνδεση και εκτέλεση εντολών με το πρωτόκολλο SSH

```
import paramiko, os, sys, random
from dotenv import load_dotenv

def main():
	load_dotenv()
	host = os.environ.get('SSH_HOSTNAME')
	user = os.environ.get('SSH_USERNAME')
	psswrd = os.environ.get('SSH_PASSWORD')

	# Setup connection and connect
	try:
		ssh_client=paramiko.SSHClient()
		ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
		ssh_client.connect(hostname=host, username=user, password=psswrd)
	except Exception as e:
		print("Connection error: " + str(e))
		sys.exit()

	# Open the ssh_orders.txt
	try:
		f = open(os.path.join(sys.path[0], "ssh_orders.txt"), "r")
	except OSError:
		print("Could not open/read file: ssh_orders.txt")
		sys.exit()

	# Create a list with the commands
	l = []
	for command in f:
		l.append(command.rstrip())
	f.close()

	# Execute a random command
	stdin,stdout,stderr = ssh_client.exec_command(random.choice(l))

	# Print stdout and stderr output
	get_results(stdin,stdout,stderr)

def get_results(stdin,stdout,stderr):
	print("stdout: ", stdout.readlines())
	print("stderr: ", stderr.readlines())

if __name__ == '__main__':
	main()
```

Κ.4 Ενδεικτικός κώδικας με τον οποίο αυτοματοποιείται η μεταφορά αρχείων με το πρωτόκολλο FTP

```
import ftplib

def main():

	try:
		# Connect to ftp1.at.proftpd.org
		ftp = ftplib.FTP('ftp1.at.proftpd.org')

		# Print the welcome message.
		print(ftp.getwelcome())

		# Login
		ftp.login()

		# The dir() method produces a directory listing and adds the
		# data to the list.

		files = []
		ftp.dir(files.append)
		print(files)

		# Get the working directory
		wdir = ftp.pwd()
		print(wdir)

		# File to download
		file_orig = 'README.MIRRORS'

		# Download the file
		with open(file_orig, 'w') as fp:
			res = ftp.retrlines('RETR ' + file_orig, fp.write)

			if not res.startswith('226 Transfer complete'):
				print('Download failed')

	except ftplib.all_errors as e:
		print('FTP error:', e)


if __name__ == '__main__':
	main()
```

Βιβλιογραφία

[1] "Web crawler - Wikipedia." https://en.wikipedia.org/wiki/Web_crawler (accessed Sep. 13, 2021).

[2] "What is a honeypot? How it protects against cyber attacks." https://searchsecurity.techtarget.com/definition/honey-pot (accessed Sep. 09, 2021).

[3] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, "Towards a Reliable Intrusion Detection Benchmark Dataset," *Softw. Netw.*, vol. 2017, no. 1, pp. 177–200, 2017, doi: 10.13052/jsn2445-9739.2017.009.

[4] "What is a Cyber Attack? | Check Point Software." https://www.checkpoint.com/cyber-hub/cyber-security/what-is-cyber-attack/ (accessed Jul. 18, 2021).

[5] "The Top 6 Industries At Risk For Cyber Attacks - RedTeam Security." https://www.redteamsecure.com/blog/the-top-6-industries-at-risk-for-cyber-attacks (accessed Jul. 18, 2021).

[6] "Cyberattack - Wikipedia." https://en.wikipedia.org/wiki/Cyberattack (accessed Jul. 18, 2021).

[7] G. Martin, P. Martin, C. Hankin, A. Darzi, and J. Kinross, "Cybersecurity and healthcare: how safe are we?," *BMJ*, vol. 358, Jul. 2017, doi: 10.1136/BMJ.J3179.

[8] S. T. Argaw *et al.*, "Cybersecurity of Hospitals: discussing the challenges and working towards mitigating the risks," *BMC Med. Informatics Decis. Mak. 2020 201*, vol. 20, no. 1, pp. 1–10, Jul. 2020, doi: 10.1186/S12911-020-01161-7.

[9] "9 Reasons Healthcare is the Biggest Target for Cyberattacks." https://swivelsecure.com/solutions/healthcare/healthcare-is-the-biggest-target-for-cyberattacks/ (accessed Jul. 18, 2021).

[10] "Attacks targeting healthcare organizations spike globally as COVID-19 cases rise again - Check Point Software." https://blog.checkpoint.com/2021/01/05/attacks-targeting-healthcare-organizations-spike-globally-as-covid-19-cases-rise-again/ (accessed Jul. 18, 2021).

[11] "Intrusion detection system - Wikipedia." https://en.wikipedia.org/wiki/Intrusion_detection_system (accessed Jul. 19, 2021).

[12] "Τι είναι IDS." http://idsportal.cs.teiath.gr/index.php/el/genika/ids (accessed Jul. 19, 2021).

[13] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges,” *Cybersecurity 2019 21*, vol. 2, no. 1, pp. 1–22, Jul. 2019, doi: 10.1186/S42400-019-0038-7.

[14] “Defining Intrusion Detection and Prevention Systems.” https://www.gartner.com/en/documents/3449317 (accessed Oct. 08, 2021).

[15] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, “Toward developing a systematic approach to generate benchmark datasets for intrusion detection,” *Comput. Secur.*, vol. 31, no. 3, pp. 357–374, 2012, doi: 10.1016/j.cose.2011.12.012.

[16] R. Damasevicius *et al.*, “Litnet-2020: An annotated real-world network flow dataset for network intrusion detection,” *Electron.*, vol. 9, no. 5, 2020, doi: 10.3390/electronics9050800.

[17] M. Odusami, S. Misra, E. Adetiba, O. Abayomi-Alli, R. Damasevicius, and R. Ahuja, “An Improved Model for Alleviating Layer Seven Distributed Denial of Service Intrusion on Webserver,” *J. Phys. Conf. Ser.*, vol. 1235, no. 1, 2019, doi: 10.1088/1742-6596/1235/1/012020.

[18] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Therón, “UGR‘16: A new dataset for the evaluation of cyclostationarity-based network IDSs,” *Comput. Secur.*, vol. 73, pp. 411–424, 2018, doi: 10.1016/j.cose.2017.11.004.

[19] J. Mchugh, “Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory,” *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, pp. 262–294, 2000, doi: 10.1145/382912.382923.

[20] C. Brown, A. Cowperthwaite, A. Hijazi, and A. Somayaji, “Analysis of the 1999 DARPA/Lincoln Laboratory IDS Evaluation Data with NetADHICT,” *IEEE Symp. Comput. Intell. Secur. Def. Appl. CISDA 2009*, no. Cisda, 2009, doi: 10.1109/CISDA.2009.5356522.

[21] “Index of /databases/kddcup99.” http://kdd.ics.uci.edu/databases/kddcup99/ (accessed Jul. 25, 2021).

[22] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” *IEEE Symp. Comput. Intell. Secur. Def. Appl. CISDA 2009*, Dec. 2009, doi: 10.1109/CISDA.2009.5356528.

[23] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” *ICISSP 2018 - Proc. 4th Int. Conf. Inf. Syst. Secur. Priv.*, vol. 2018-Janua, no. Cic, pp. 108–116, 2018, doi: 10.5220/0006639801080116.

[24] “LBNL/ICSI Enterprise Tracing Project - Project Overview.” http://www.icir.org/enterprise-tracing/ (accessed Sep. 09, 2021).

[25] “(PDF) Toward instrumenting network warfare competitions to generate labeled datasets.” https://www.researchgate.net/publication/234802396_Toward_instrumenting_network_warfare_competitions_to_generate_labeled_datasets (accessed Sep. 09, 2021).

[26] A. Sperotto, R. Sadre, F. Van Vliet, and A. Pras, “A labeled data set for flow-based intrusion detection,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5843 LNCS, pp. 39–50, 2009, doi: 10.1007/978-3-642-04968-2_4.

[27] “Home Page - UMass Trace Repository.” http://traces.cs.umass.edu/ (accessed Sep. 10, 2021).

[28] S. Prusty, B. N. Levine, and M. Liberatore, “Forensic investigation of the OneSwarm anonymous filesharing system,” *Proc. ACM Conf. Comput. Commun. Secur.*, pp. 201–213, 2011, doi: 10.1145/2046707.2046731.

[29] J. O. Nehinbe, “A critical evaluation of datasets for investigating IDSs and IPSs researches,” *Proc. 2011, 10th IEEE Int. Conf. Cybern. Intell. Syst. CIS 2011*, pp. 92–97, 2011, doi: 10.1109/CIS.2011.6169141.

[30] G. Creech and J. Hu, “Generation of a new IDS test dataset: Time to retire the KDD collection,” *IEEE Wirel. Commun. Netw. Conf. WCNC*, pp. 4487–4492, 2013, doi: 10.1109/WCNC.2013.6555301.

[31] M. Xie and J. Hu, “Evaluating host-based anomaly detection systems: A preliminary analysis of ADFA-LD,” *Proc. 2013 6th Int. Congr. Image Signal Process. CISP 2013*, vol. 3, pp. 1711–1716, 2013, doi: 10.1109/CISP.2013.6743952.

[32] N. Moustafa, J. S.-2015 military communications and, and undefined 2015, “UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” *ieeexplore.ieee.org*, doi: 10.1109/MilCIS.2015.7348942.

[33] “Introduction to Cisco IOS ® NetFlow-A Technical Overview,” 1992.

[34] C. J. Date and E. F. Codd, “The relational and network approaches: Comparison of the application programming interfaces,” *Proc. ACM SIGMOD Int. Conf. Manag. Data*, pp. 83–113, Jan. 1975, doi: 10.1145/800297.811532.

[35] “Architectural Styles and the Design of Network-based Software Architectures.” https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm (accessed Oct. 02, 2021).

[36] B. Jin, S. Sahni, and A. Shevat, “Designing Web APIs,” p. 207, 2018.

[37] “RESTful Web APIs 1st edition by Richardson, Leonard, Amundsen, Mike, Ruby, Sam (2013) Paperback: Richardson, Leonard, Amundsen, Mike, Ruby, Sam:

Amazon.com: Books.” https://www.amazon.com/RESTful-Richardson-Leonard-Amundsen-Paperback/dp/B011DBNFVK (accessed Oct. 02, 2021).

[38] “Representational state transfer - Wikipedia.” https://en.wikipedia.org/wiki/Representational_state_transfer (accessed Oct. 02, 2021).

[39] O. Ferreira, *Semantic Web Services: A RESTful Approach*. IADIS, 2009.

[40] L. Richardson and M. Amundsen, *RESTful Web APIs*. O’Reilly Media, 2013.

[41] “What is REST API,” *RESTful API Tutor.*, Accessed: Oct. 02, 2021. [Online]. Available: http://restfulapi.net/.

[42] M. Massé, “REST-API-Design,” *O’Reilly*, p. 114, 2012.

[43] “What is Swagger.” https://swagger.io/docs/specification/2-0/what-is-swagger/ (accessed Oct. 02, 2021).

[44] “Swagger (software) - Wikipedia.” https://en.wikipedia.org/wiki/Swagger_(software) (accessed Oct. 02, 2021).

[45] “Welcome to Python.org.” https://www.python.org/ (accessed Oct. 01, 2021).

[46] “InterProjektWiki: NetFlow.” https://pliki.ip-sa.pl/wiki/Wiki.jsp?page=NetFlow (accessed Sep. 24, 2021).

[47] “Selenium (software) - Wikipedia.” https://en.wikipedia.org/wiki/Selenium_(software) (accessed Oct. 01, 2021).

[48] “Selenium.” https://www.selenium.dev/ (accessed Oct. 01, 2021).

[49] “SkPy.docs/index.rst at master · Terrance/SkPy.docs.” https://github.com/Terrance/SkPy.docs/blob/master/index.rst (accessed Oct. 01, 2021).

[50] “Welcome to Paramiko’s documentation! — Paramiko documentation.” http://docs.paramiko.org/en/stable/index.html (accessed Oct. 01, 2021).

[51] “Paramiko- How to SSH and transfer files with python | by Mokgadi Rasekgala | Medium.” https://medium.com/@keagileageek/paramiko-how-to-ssh-and-file-transfers-with-python-75766179de73 (accessed Oct. 01, 2021).

[52] “Python FTP programming - Python ftplib.” https://zetcode.com/python/ftp/ (accessed Oct. 01, 2021).